

On Generalized Bellman Equations and Temporal-Difference Learning*

Huizhen Yu[†], A. Rupam Mahmood[‡] and Richard S. Sutton[§]

RLAI Lab, Department of Computing Science
University of Alberta, Canada

Abstract

We consider off-policy temporal-difference (TD) learning in discounted Markov decision processes, where the goal is to evaluate a policy in a model-free way by using observations of a state process generated without executing the policy. To curb the high variance issue in off-policy TD learning, we propose a new scheme of setting the λ -parameters of TD, based on generalized Bellman equations. Our scheme is to set λ according to the eligibility trace iterates calculated in TD, thereby easily keeping these traces in a desired bounded range. Compared to prior works, this scheme is more direct and flexible, and allows much larger λ values for off-policy TD learning with bounded traces. Using Markov chain theory, we prove the ergodicity of the joint state-trace process under nonrestrictive conditions, and we show that associated with our scheme is a generalized Bellman equation (for the policy to be evaluated) that depends on both the evolution of λ and the unique invariant probability measure of the state-trace process. These results not only lead immediately to a characterization of the convergence behavior of least-squares based implementation of our scheme, but also prepare the ground for further analysis of gradient-based implementations.

Keywords: Markov decision process; approximate policy evaluation; generalized Bellman equation; reinforcement learning; temporal-difference method; Markov chain; randomized stopping time

*This research was supported by a grant from Alberta Innovates—Technology Futures. An abridged version of this article is to appear at *The 30th Canadian Conference on Artificial Intelligence*, May, 2017.

[†]janey.hzyu@gmail.com

[‡]ashique@ualberta.ca

[§]rsutton@ualberta.ca

Contents

1	Introduction	3
2	Off-Policy TD Learning with Bounded Traces	3
2.1	Preliminaries	3
2.2	Our Choice of λ	5
2.3	Ergodicity Result	7
3	Generalized Bellman Equations	11
3.1	Randomized Stopping Times	11
3.2	Bellman Equation for the Proposed TD Learning Scheme	13
3.3	Proofs of Theorem 3.2 and Corollary 3.1	16
4	Numerical Study	18
4.1	Behavior of Traces	18
4.1.1	Traces for Constant λ	19
4.1.2	Traces with Evolving λ	20
4.2	LSTD with Evolving λ	21
4.2.1	A Toy Problem	21
4.2.2	Mountain Car Problem	23
5	Conclusion	28
	Appendices	29
	Appendix A: Proof of Theorem 3.1	29
	Appendix B: Oblique Projection Viewpoint and Error Bound for TD	31
	B.1 Solutions of TD as Oblique Projections of the Value Function	31
	B.2 Approximation Error Bound	33
	References	34

1 Introduction

We consider off-policy temporal-difference (TD) learning in discounted Markov decision processes (MDPs), where the goal is to evaluate a policy in a model-free way by using observations of a state process generated without executing the policy. Off-policy learning is an important part of the reinforcement learning methodology [30] and has been studied in the areas of operations research and machine learning (see e.g., [4, 6, 7, 11, 12, 13, 14, 20, 21, 23, 36]). Available algorithms, however, tend to have very high variances due to the use of importance sampling, an issue that limits their applicability in practice. The purpose of this paper is to introduce a new TD learning scheme that can help address this problem.

Our work is motivated by the recently proposed Retrace algorithm [18] and ABQ algorithm [15], and by the Tree-Backup algorithm [21] that existed earlier. These algorithms, as explained in [15], all try to use the λ -parameters of TD to curb the high variance issue in off-policy learning. In this paper we propose a new scheme of setting the λ -parameters of TD, based on generalized Bellman equations. Our scheme is to set λ according to the eligibility trace iterates calculated in TD, thereby easily keeping those traces in a desired bounded range. Compared to the previous works, this is a direct way to bound the traces in TD, and it is also more flexible, allowing much larger λ values for off-policy learning.

Regarding generalized Bellman equations, they are a powerful tool. In classic MDP theory they have been used in some intricate optimality analyses. Their computational use, however, seems to emerge primarily in the field of reinforcement learning (see [29], [1, Chap. 5.3] and [35] for related early and recent research). Like the earlier works [15, 18, 21, 35, 41], our work aims to employ this tool to make off-policy learning more efficient.

Our analyses of the new TD learning scheme will focus on its theoretical side. Using Markov chain theory, we prove the ergodicity of the joint state and trace process under nonrestrictive conditions (see Theorem 2.1), and we show that associated with our scheme is a generalized Bellman equation (for the policy to be evaluated) that depends on both the evolution of λ and the unique invariant probability measure of the state-trace process (see Theorem 3.2 and Corollary 3.1). These results not only lead immediately to a characterization of the convergence behavior of least-squares based implementation of our scheme (see Corollary 2.1 and Remark 3.2), but also prepare the ground for further analysis of gradient-based implementations (see Remark 3.4).

In addition to the theoretical study, we also present the results from a preliminary numerical study that compares several ways of setting λ for the least-squares based off-policy algorithm. The results demonstrate the advantages of the proposed new scheme with its greater flexibility.

The rest of the paper is organized as follows. In Section 2, after a brief background introduction, we present our scheme of TD learning with bounded traces, and we establish the ergodicity of the joint state-trace process. In Section 3, we first discuss generalized Bellman operators associated with randomized stopping times, and we then derive the generalized Bellman equation associated with our scheme. In Section 4, we present the experimental results on the least-squares based implementation of our scheme. Appendices A-B include a proof for generalized Bellman operators and materials about approximation properties of TD solutions that are too long to include in the main text.

2 Off-Policy TD Learning with Bounded Traces

2.1 Preliminaries

The off-policy learning problem we consider in this paper concerns two Markov chains on a finite state space $\mathcal{S} = \{1, \dots, N\}$. The first chain has transition matrix P , and the second P° . Whatever physical mechanisms that induce the two chains shall be denoted by π and π° , and referred to as the

target policy and behavior policy, respectively. The second Markov chain we can observe; however, it is the system performance for the first Markov chain that we want to evaluate.

Specifically, we consider a one-stage reward function $r_\pi : \mathcal{S} \rightarrow \mathbb{R}$ and an associated discounted total reward criterion with state-dependent discount factors $\gamma(s) \in [0, 1], s \in \mathcal{S}$. Let Γ denote the $N \times N$ diagonal matrix with diagonal entries $\gamma(s)$. We assume that P and P^o satisfy the following conditions:

Condition 2.1 (Conditions on the target and behavior policies).

- (i) P is such that the inverse $(I - P\Gamma)^{-1}$ exists, and
- (ii) P^o is such that for all $s, s' \in \mathcal{S}$, $P_{ss'}^o = 0 \Rightarrow P_{ss'} = 0$, and moreover, P^o is irreducible.

The performance of π is defined as the expected discounted total rewards for each initial state $s \in \mathcal{S}$:

$$v_\pi(s) := \mathbb{E}_s^\pi [r_\pi(S_0) + \sum_{t=1}^\infty \gamma(S_1) \gamma(S_2) \cdots \gamma(S_t) \cdot r_\pi(S_t)], \quad (2.1)$$

where the notation \mathbb{E}_s^π means that the expectation is taken with respect to (w.r.t.) the Markov chain $\{S_t\}$ starting from $S_0 = s$ and induced by π (i.e., with transition matrix P). The function v_π is well-defined under Condition 2.1(i). It is called the *value function* of π , and by standard MDP theory (see e.g., [22]), we can write it in matrix/vector notation as

$$v_\pi = r_\pi + P\Gamma v_\pi, \quad \text{i.e.,} \quad v_\pi = (I - P\Gamma)^{-1} r_\pi.$$

The first equation above is known as the Bellman equation (or dynamic programming equation) for a stationary policy (cf. Footnote 2).

We compute an approximation of v_π of the form $v(s) = \phi(s)^\top \theta$, $s \in \mathcal{S}$, where $\theta \in \mathbb{R}^n$ is a parameter vector and $\phi(s)$ is an n -dimensional feature representation for each state s ($\phi(s), \theta$ are column vectors and $^\top$ stands for transpose). Data available for this computation are:

- (i) the Markov chain $\{S_t\}$ with transition matrix P^o generated by π^o , and
- (ii) rewards $R_t = r(S_t, S_{t+1})$ associated with state transitions, where the function r relates to $r_\pi(s)$ as $r_\pi(s) = \mathbb{E}_s^\pi [r(s, S_1)]$ for all $s \in \mathcal{S}$.¹

To find a suitable parameter θ for the approximation $\phi(s)^\top \theta$, we use the off-policy TD learning scheme. Define $\rho(s, s') = P_{ss'}/P_{ss'}^o$ (the importance sampling ratio),² and write

$$\rho_t = \rho(S_t, S_{t+1}), \quad \gamma_t = \gamma(S_t).$$

Given an initial $e_0 \in \mathbb{R}^n$, for each $t \geq 1$, the eligibility trace vector $e_t \in \mathbb{R}^n$ and the scalar temporal-difference term $\delta_t(v)$ for any approximate value function $v : \mathcal{S} \rightarrow \mathbb{R}$ are calculated according to

$$e_t = \lambda_t \gamma_t \rho_{t-1} e_{t-1} + \phi(S_t), \quad (2.2)$$

$$\delta_t(v) = \rho_t (R_t + \gamma_{t+1} v(S_{t+1}) - v(S_t)). \quad (2.3)$$

¹One can add to R_t a zero-mean finite-variance noise term. This makes little difference to our analyses, so we have left it out for notational simplicity.

²Our problem formulation entails both value function and state-action value function estimation for a stationary policy in the standard MDP context. In these applications, it is the state-action space of the MDP that corresponds to the state space \mathcal{S} here. In particular, for value function estimation, S_t here corresponds to the pair of previous action and current state in the MDP, whereas for state-value function estimation, S_t here corresponds to the current state-action pair in the MDP. The ratio $\rho(s, s') = P_{ss'}/P_{ss'}^o$ then comes out as the ratio of action probabilities under π and π^o , the same as what appears in most of the off-policy learning literature. For the details of these correspondences, see [36, Examples 2.1, 2.2]. The third application is in a simulation context where P^o corresponds to a simulated system and both P^o, P are known so that the ratio $\rho(s, s')$ is available. Such simulations are useful, for example, in studying system performance under perturbations, and in speeding up the computation when assessing the impacts of events that are rare under the dynamics P .

Here $\lambda_t \in [0, 1], t \geq 1$, are important parameters in TD learning, the choice of which we shall elaborate on shortly.

Using e_t and δ_t , a number of algorithms can be formed to generate a sequence of parameters θ_t for approximate value functions. One such algorithm is LSTD [3, 36], which obtains θ_t by solving the linear equation for $\theta \in \mathbb{R}^n$,

$$\frac{1}{t} \sum_{k=0}^{t-1} e_k \delta_k(v) = 0, \quad v = \Phi\theta \quad (2.4)$$

(if it admits a solution), where Φ is a matrix with row vectors $\phi(s)^\top, s \in \mathcal{S}$. LSTD updates the equation (2.4) iteratively by incorporating one by one the observation of (S_t, S_{t+1}, R_t) at each state transition. We will discuss primarily this algorithm in the paper, as its behavior can be characterized directly using our subsequent analyses of the joint state-trace process.

As mentioned earlier, our analyses will also provide bases for analyzing other gradient-based TD algorithms [12, 13, 31, 33] using stochastic approximation theory [2, 9, 10]. However, due to its complexity, this subject is better to be treated separately, not in the present paper.

2.2 Our Choice of λ

We now come to the choices of λ_t in the trace iterates (2.2). For TD with function approximation, one often lets λ_t be a constant or a function of S_t [28, 30, 34]. If neither the behavior policy nor the λ_t 's are further constrained, $\{e_t\}$ can have unbounded variances and is also unbounded in many natural situations (see e.g., [36, Section 3.1]), and this makes off-policy TD learning challenging.³ If we let the behavior policy to be close enough to the target policy so that $P^o \approx P$, then variance can be reduced, but it is not a satisfactory solution, for the applicability of off-policy learning would be seriously limited.

Without restricting the behavior policy, the two recent works [15, 18] (as well as the closely related early work [21]) exploit state-dependent λ 's to control variance. Their choices of λ_t are such that $\lambda_t \gamma_t \rho_{t-1} < 1$ for all t , so that the trace iterates e_t are made bounded, which can help reduce the variance of the iterates.

Our proposal, motivated by these prior works, is to set λ_t according to e_{t-1} directly, so that we can keep e_t in a desired range straightforwardly and at the same time, allow a much larger range of values for the λ -parameters. As a simple example, if we use λ_t to scale the vector $\gamma_t \rho_{t-1} e_{t-1}$ to be within a ball with some given radius, then we keep e_t bounded always.

In the rest of this paper, we shall focus on analyzing the iteration (2.2) with a particular choice of λ_t of the kind just mentioned. We want to be more general than the preceding simple example. However, we also want to retain certain Markovian properties that are very useful for convergence analysis. This leads us to consider λ_t *being a certain function of the previous trace and past states*. More specifically, we will let λ_t be a function of the previous trace and a certain memory state that is a summary of the states observed so far, and the formulation is as follows.

Denote the memory state at time t by y_t . For simplicity, we assume that y_t can only take values from a finite set \mathcal{M} , and its evolution is Markovian: $y_t = g(y_{t-1}, S_t)$ for some given function g . The joint process $\{(S_t, y_t)\}$ is then a simple finite-state Markov chain. Each y_t is a function of the history (S_0, \dots, S_t) and y_0 . We further require, besides the irreducibility of $\{S_t\}$ (cf. Condition 2.1(ii)), that⁴

Condition 2.2 (Evolution of memory states). *Under the behavior policy π^o , the Markov chain $\{(S_t, y_t)\}$ on $\mathcal{S} \times \mathcal{M}$ has a single recurrent class.*

Thus we let y_t and λ_t evolve as

$$y_t = g(y_{t-1}, S_t), \quad \lambda_t = \lambda(y_t, e_{t-1}) \quad (2.5)$$

³Asymptotic convergence is still ensured, however, for several algorithms [36, 37, 39], thanks partly to a powerful law of large numbers for stationary processes.

⁴These conditions are nonrestrictive. If the Markov chains have multiple recurrent classes, each recurrent class can be treated separately using the same arguments.

where $\lambda : \mathcal{M} \times \mathbb{R}^n \rightarrow [0, 1]$. We require the function λ to satisfy two conditions.

Condition 2.3 (Conditions for λ). *For some norm $\|\cdot\|$ on \mathbb{R}^n , the following hold for each memory state $y \in \mathcal{M}$:*

- (i) *For any $e, e' \in \mathbb{R}^n$, $\|\lambda(y, e)e - \lambda(y, e')e'\| \leq \|e - e'\|$.*
- (ii) *For some constant C_y , $\|\gamma(s')\rho(s, s') \cdot \lambda(y, e)e\| \leq C_y$ for all $e \in \mathbb{R}^n$ and all possible state transitions (s, s') that can lead to the memory state y .*

In the above, the second condition is to restrict $\{e_t\}$ in a desired range (as it makes $\|e_t\| \leq \max_{y \in \mathcal{M}} C_y + \max_{s \in \mathcal{S}} \|\phi(s)\|$). The first condition is to ensure that the traces e_t jointly with (S_t, y_t) form a Markov chain with nice properties, and it plays a key role in the analysis to be given in the next subsection. We shall defer a further discussion on the technical roles of these conditions to the end of the next subsection. Here let us discuss a few simple examples of choosing λ that satisfy Condition 2.3.

Example 2.1. We consider again the simple scaling example mentioned earlier and describe it using the terminologies just introduced. In this example, we can let $y_t = (S_{t-1}, S_t)$, and for each $y = (s, s')$, we define $\lambda(y, \cdot)$ so that the vector $\gamma(s')\rho(s, s')e$ is scaled back whenever it is outside the Euclidean ball with radius $C_{ss'}$:

$$\lambda(y, e) = \begin{cases} 1 & \text{if } \gamma(s')\rho(s, s')\|e\|_2 \leq C_{ss'}; \\ \frac{C_{ss'}}{\gamma(s')\rho(s, s')\|e\|_2} & \text{otherwise.} \end{cases} \quad (2.6)$$

Condition 2.3(i) is satisfied because for $y = (s, s')$ with $\gamma(s')\rho(s, s') = 0$, $\lambda(y, e)e = e$, whereas for $y = (s, s')$ with $\gamma(s')\rho(s, s') \neq 0$, $\lambda(y, e)e$ is simply the Euclidean projection of e onto the ball with radius $C_{ss'}/(\gamma(s')\rho(s, s'))$ and is therefore Lipschitz continuous in e with modulus 1 w.r.t. $\|\cdot\|_2$. Corresponding to (2.6), the update of e_t becomes

$$e_t = \begin{cases} \gamma_t \rho_{t-1} e_{t-1} + \phi(S_t) & \text{if } \gamma_t \rho_{t-1} \|e_{t-1}\|_2 \leq C_{S_{t-1}S_t}; \\ C_{S_{t-1}S_t} \cdot \frac{e_{t-1}}{\|e_{t-1}\|_2} + \phi(S_t) & \text{otherwise.} \end{cases} \quad (2.7)$$

Note that this scheme of setting λ encourages the use of large λ_t : $\lambda_t = 1$ will be chosen whenever possible. A variation of the scheme is to multiply the r.h.s. of (2.6) by another factor $\beta_{ss'} \in [0, 1]$, so that λ_t can be at most $\beta_{S_{t-1}S_t}$. In particular, one such variation is to simply multiply the r.h.s. of (2.6) by a memory-independent constant $\beta \in (0, 1)$ so that $\lambda_t \leq \beta < 1$ always. \square

Example 2.2. The Retrace algorithm [18] modifies the trace updates in off-policy TD learning by truncating the importance sampling ratios by 1. In particular, for the off-policy TD(λ) algorithm with a constant $\lambda = \beta \in (0, 1]$, Retrace modifies the trace updates to be

$$e_t = \beta \gamma_t \cdot \min\{1, \rho_{t-1}\} \cdot e_{t-1} + \phi(S_t). \quad (2.8)$$

As pointed out in [15], to retain the original interpretation of λ as a bootstrapping parameter in TD learning, we can rewrite the above update rule of Retrace equivalently as

$$e_t = \lambda_t \gamma_t \rho_{t-1} e_{t-1} + \phi(S_t) \quad \text{for } \lambda_t = \beta \cdot \frac{\min\{1, \rho_{t-1}\}}{\rho_{t-1}} \quad (\text{with } 0/0 = 0). \quad (2.9)$$

Each λ_t here is a function of (S_{t-1}, S_t) only and does not depend on e_{t-1} , so this choice of λ -parameters automatically satisfies Condition 2.3(i) with the memory states being $y_t = (S_{t-1}, S_t)$. When the discount factors $\gamma(s)$ are all strictly less than 1, $\|e_t\|$ for all t are bounded by a deterministic constant that depends on the initial e_0 . Then for each initial e_0 , Retrace's choice of λ coincides with a choice in our framework, since the C -parameters in Condition 2.3(ii) can be made vacuously large

so that the condition is satisfied by all the traces e_t that could be encountered by Retrace. Thus in this case our framework for choosing λ effectively encompasses the particular choice used by Retrace.

One can make variations on Retrace’s trace update rule. For example, instead of truncating each importance sampling ratio $\rho(s, s')$ by 1, one can truncate it by a constant $K_{ss'} \geq 1$, and then use a scaling scheme similar to Example 2.1 to bound the traces. The simplest such variation is to choose two memory-independent positive constants K and C , and replace the definition of λ_t in (2.9) by the following: with $\tilde{\lambda}_t = \frac{\min\{K, \rho_{t-1}\}}{\rho_{t-1}}$ (where we treat $0/0 = 0$),

$$\lambda_t = \begin{cases} \beta \tilde{\lambda}_t & \text{if } \tilde{\lambda}_t \gamma_t \rho_{t-1} \|e_{t-1}\|_2 \leq C; \\ \beta \tilde{\lambda}_t \cdot \frac{C}{\tilde{\lambda}_t \gamma_t \rho_{t-1} \|e_{t-1}\|_2} & \text{otherwise.} \end{cases} \quad (2.10)$$

Correspondingly, instead of (2.8), the update of e_t becomes

$$e_t = \begin{cases} \beta \gamma_t \cdot \min\{K, \rho_{t-1}\} \cdot e_{t-1} + \phi(S_t) & \text{if } \gamma_t \cdot \min\{K, \rho_{t-1}\} \cdot \|e_{t-1}\|_2 \leq C; \\ \beta C \cdot \frac{e_{t-1}}{\|e_{t-1}\|_2} + \phi(S_t) & \text{otherwise.} \end{cases} \quad (2.11)$$

These variations of Retrace are similar to Example 2.1 and satisfy Condition 2.3.⁵ \square

2.3 Ergodicity Result

The properties of the joint state-trace process $\{(S_t, y_t, e_t)\}$ are important for understanding and characterizing the behavior of the proposed TD learning scheme. We study them in this subsection. Most importantly, we shall establish the ergodicity of the state-trace process. The result will be useful in convergence analysis of several associated TD algorithms, although in this paper we discuss only the LSTD algorithm. In the next section we will also use the ergodicity result when we relate the LSTD equation (2.4) to a generalized Bellman equation for the target policy, which will then make the meaning of the LSTD solutions clear.

As a side note, one can introduce nonnegative coefficients $i(y)$ for memory states y to weight the state features (similarly to the use of “interest” weights in the ETD algorithm [32]) and update e_t according to

$$e_t = \lambda_t \gamma_t \rho_{t-1} e_{t-1} + i(y_t) \phi(S_t). \quad (2.12)$$

The results given below apply to this update rule as well.

Let us start with two basic properties of $\{(S_t, y_t, e_t)\}$ that follow directly from our choice of the λ function:

- (i) By Condition 2.3(i), for each y , $\lambda(y, e)e$ is a continuous function of e , and thus e_t depends continuously on e_{t-1} . This, together with the finiteness of $\mathcal{S} \times \mathcal{M}$, ensures that $\{(S_t, y_t, e_t)\}$ is a weak Feller Markov chain.⁶
- (ii) Then, by a property of weak Feller Markov chains [17, Theorem 12.1.2(ii)], the boundedness of $\{e_t\}$ ensured by Condition 2.3(ii) implies that $\{(S_t, y_t, e_t)\}$ has at least one invariant probability measure.

The third property, given in the lemma below, concerns the behavior of $\{e_t\}$ for different initial e_0 . It is an important implication of Condition 2.3(i); actually, it is our purpose of introducing the condition 2.3(i) in the first place. In the lemma, $\xrightarrow{a.s.}$ stands for “converges almost surely to.”

⁵To see this, let the memory states be $y_t = (S_{t-1}, S_t)$. For each $y = (s, s')$, let $\lambda(y, e)$ be defined according to (2.10), and let C_y in Condition 2.3(ii) be $C_{ss'} = \frac{\rho(s, s')}{\min\{K, \rho(s, s')\}} \cdot C$ (treat $0/0 = 0$). Then note that $\|\lambda(y, e)e - \lambda(y, e')e'\|_2 \leq \beta \min\{\frac{K}{\rho(s, s')}, 1\} \cdot \|e - e'\|_2 \leq \beta \|e - e'\|_2$.

⁶This means that for any bounded continuous function f on $\mathcal{S} \times \mathcal{M} \times \mathbb{R}^n$ (endowed with the usual topology), with $X_t = (S_t, y_t, e_t)$, $\mathbb{E}[f(X_1) \mid X_0 = x]$ is a continuous function of x [17, Prop. 6.1.1].

Lemma 2.1. *Let $\{e_t\}$ and $\{\hat{e}_t\}$ be generated by the iteration (2.2) and (2.5), using the same trajectory of states $\{S_t\}$ and initial y_0 , but with different initial e_0 and \hat{e}_0 , respectively. Then under Conditions 2.1(i) and 2.3(i), $e_t - \hat{e}_t \xrightarrow{a.s.} 0$.*

Proof. The proof is similar to that of [36, Lemma 3.2]. Let $\Delta_t = \|e_t - \hat{e}_t\|$, and let \mathcal{F}_t denote the σ -algebra generated by $S_k, k \leq t$. Note that under our assumption, in the generation of $\{e_t\}, \{\hat{e}_t\}$, the states $\{S_t\}$ and the memory states $\{y_t\}$ are the same, but the λ 's are different. Let us denote them by $\{\lambda_t\}$ and $\{\hat{\lambda}_t\}$ for the two trace sequences, respectively. Then by (2.2), $e_t - \hat{e}_t = \gamma_t \rho_{t-1} (\lambda_t e_{t-1} - \hat{\lambda}_t \hat{e}_{t-1})$, and by Condition 2.3(i), $\|\lambda_t e_{t-1} - \hat{\lambda}_t \hat{e}_{t-1}\| \leq \|e_{t-1} - \hat{e}_{t-1}\|$. Hence $\|e_t - \hat{e}_t\| \leq \gamma_t \rho_{t-1} \|e_{t-1} - \hat{e}_{t-1}\|$, so $\mathbb{E}[\Delta_t | \mathcal{F}_{t-1}] \leq \mathbb{E}[\gamma_t \rho_{t-1} | \mathcal{F}_{t-1}] \cdot \Delta_{t-1} \leq \Delta_{t-1}$. This shows $\{(\Delta_t, \mathcal{F}_t)\}$ is a nonnegative supermartingale. By the supermartingale convergence theorem [5, Theorem 10.5.7 and Lemma 4.3.3], $\{\Delta_t\}$ converges a.s. to a nonnegative random variable Δ_∞ with $\mathbb{E}[\Delta_\infty] \leq \liminf_{t \rightarrow \infty} \mathbb{E}[\Delta_t]$. From the inequality $\|e_t - \hat{e}_t\| \leq \gamma_t \rho_{t-1} \|e_{t-1} - \hat{e}_{t-1}\|$ for all t , we have $\Delta_t \leq \Delta_0 \cdot \prod_{k=1}^t \gamma_k \rho_{k-1}$, from which a direct calculation shows $\mathbb{E}[\Delta_t] \leq \Delta_0 \cdot \mathbf{1}^\top (P\Gamma)^t \mathbf{1}$ where $\mathbf{1}$ denotes the n -dimensional vector of all 1's. As $t \rightarrow \infty$, $(P_\pi \Gamma)^t$ converges to the zero matrix under Condition 2.1(i). Therefore, $\liminf_{t \rightarrow \infty} \mathbb{E}[\Delta_t] = 0$ and consequently, we must have $\Delta_\infty = 0$ a.s., i.e., $\Delta_t \xrightarrow{a.s.} 0$. \square

We use Lemma 2.1 and ergodicity properties of weak Feller Markov chains [16] to prove the ergodicity theorem below. A direct application to LSTD will be discussed immediately after the theorem, before we give its proof.

To state the result, we need some terminology and notation. For $\{(S_t, y_t, e_t)\}$ starting from the initial condition $x = (s, y, e)$, we write \mathbf{P}_x for its probability distribution, and we write “ \mathbf{P}_x -a.s.” for “almost surely with respect to \mathbf{P}_x .” Its *occupation probability measures* are denoted by $\{\mu_{x,t}\}$, and they are random probability measures on $\mathcal{S} \times \mathcal{M} \times \mathbb{R}^n$ given by

$$\mu_{x,t}(D) := \frac{1}{t} \sum_{k=0}^{t-1} \mathbb{1}((S_k, y_k, e_k) \in D) \quad \forall \text{ Borel sets } D \subset \mathcal{S} \times \mathcal{M} \times \mathbb{R}^n,$$

where $\mathbb{1}(\cdot)$ is the indicator function. We are interested in the asymptotic convergence of these occupation probability measures in the sense of *weak convergence*: for probability measures $\{\mu_t\}$ and μ on a metric space, $\{\mu_t\}$ converges weakly to μ if $\int f d\mu_t \rightarrow \int f d\mu$ as $t \rightarrow \infty$, for every bounded continuous function f .

We shall also consider the Markov chain $\{(S_t, S_{t+1}, y_t, e_t)\}$, whose occupation probability measures are defined likewise. Although essentially the same as $\{(S_t, y_t, e_t)\}$, this chain is more convenient for applying our ergodicity result to TD algorithms because the temporal-difference term $\delta_t(v)$ involves (S_t, S_{t+1}, e_t) . Regarding invariant probability measures of the two Markov chains, obviously, if ζ is an invariant probability measure of $\{(S_t, y_t, e_t)\}$, then an invariable probability measure of $\{(S_t, S_{t+1}, y_t, e_t)\}$ is the probability measure ζ_1 composed from the marginal ζ and the conditional distribution of S_1 given (S_0, y_0, e_0) specified by P^o ; i.e.,

$$\zeta_1(D) = \int \sum_{s' \in \mathcal{S}} P_{ss'}^o \mathbb{1}((s, s', y, e) \in D) \zeta(d(s, y, e)) \quad \forall \text{ Borel sets } D \subset \mathcal{S}^2 \times \mathcal{M} \times \mathbb{R}^n. \quad (2.13)$$

Theorem 2.1. *Let Conditions 2.1-2.3 hold. Then $\{(S_t, y_t, e_t)\}$ is a weak Feller Markov chain and has a unique invariant probability measure ζ . For each initial condition $(S_0, y_0, e_0) = (s, y, e) =: x$, the occupation probability measures $\{\mu_{x,t}\}$ converge weakly to ζ , \mathbf{P}_x -a.s.*

Likewise, the same holds for $\{(S_t, S_{t+1}, y_t, e_t)\}$, whose unique invariant probability measure is as given in (2.13).

If the initial distribution of (S_0, y_0, e_0) is ζ , the state-trace process $\{(S_t, y_t, e_t)\}$ is stationary. Let \mathbb{E}_ζ denote expectation w.r.t. to this stationary process.

Consider now the sequence of equations in v , $\frac{1}{t} \sum_{k=0}^{t-1} e_k \delta_k(v) = 0$, appeared in (2.4) for LSTD. From the definition (2.3) of $\delta_t(v)$,

$$\delta_t(v) = \rho_t (R_t + \gamma_{t+1} v(S_{t+1}) - v(S_t)),$$

we see that for fixed v , every $e_k \delta_k(v)$ can be expressed as $f(S_k, S_{k+1}, e_k)$ for a continuous function f . Since the traces and hence the entire process lie in a bounded set under Condition 2.3(ii), the weak convergence of the occupation probabilities measures of $\{(S_t, S_{t+1}, y_t, e_t)\}$ shown by Theorem 2.1 implies that this sequence of equations has an asymptotic limit that can be expressed in terms of the stationary state-trace process as follows.

Corollary 2.1. *Let Conditions 2.1-2.3 hold. Then for each initial condition of (S_0, y_0, e_0) , almost surely, the sequence of linear equations in v , $\frac{1}{t} \sum_{k=0}^{t-1} e_k \delta_k(v) = 0$, tends asymptotically to $\mathbb{E}_\zeta[e_0 \delta_0(v)] = 0$ (also a linear equation in v), in the sense that the random coefficients in the former equations converge to the corresponding coefficients in the latter equation as $t \rightarrow \infty$.*

In the rest of this section we prove Theorem 2.1. Broadly speaking, the line of argument is as follows: We first prove the weak convergence of occupation probability measures to the same invariant probability measure, for each initial condition. This will in turn imply the uniqueness of the invariant probability measure.

After the proof we will first comment in Remark 2.1 on the differences between our proof and that of a similar result in the previous work [36]. We will then comment in Remark 2.2 about the technical roles of the Condition 2.3 on the choice of λ and whether some part of that condition can be relaxed.

Proof of Theorem 2.1. As we discussed before Lemma 2.1, under Conditions 2.3, $\{(S_t, y_t, e_t)\}$ is weak Feller and has at least one invariant probability measure ζ . Then, by [16, Prop. 4.1], there exists a set $D \subset \mathcal{S} \times \mathcal{M} \times \mathbb{R}^n$ with ζ -measure 1 such that for each initial condition $x = (s, y, e) \in D$, the occupation probability measures $\{\mu_{x,t}\}$ converge weakly, \mathbf{P}_x -a.s., to an invariant probability measure μ_x that depends only on the initial condition x . To prove the theorem using this result, we need to show that (i) all these $\{\mu_x \mid x \in D\}$ are the same invariant probability measure, and (ii) for all $x \notin D$, $\{\mu_{x,t}\}$ has the same weak convergence property.

To this end, we first consider an arbitrary pair (s, y_s) in the recurrent class of $\{(S_t, y_t)\}$ (cf. Condition 2.2). Let us show that for all initial conditions $x \in \{(s, y_s, e) \mid e \in \mathbb{R}^n\}$, $\{\mu_{x,t}\}$ converges weakly to the same invariant probability measure, almost surely.

Since the finite-state Markov chain $\{(S_t, y_t)\}$ has a single recurrent class (Condition 2.2) and its evolution is not affected by $\{e_t\}$, the marginal of ζ on $\mathcal{S} \times \mathcal{M}$ coincides with the unique invariant probability distribution of $\{(S_t, y_t)\}$. So the fact that $\zeta(D) = 1$ and (s, y_s) is a recurrent state of $\{(S_t, y_t)\}$ implies that there exists some \hat{e} with $(s, y_s, \hat{e}) \in D$. For the initial condition $\hat{x} = (s, y_s, \hat{e})$, by the result of [16] mentioned earlier, $\{\mu_{\hat{x},t}\}$ converges weakly to $\mu_{\hat{x}}$, almost surely.

Now consider $x = (s, y_s, e)$ for an arbitrary $e \in \mathbb{R}^n$. Generate iterates $\{\hat{e}_t\}$ and $\{e_t\}$ according to (2.2), using the same trajectory $\{(S_t, y_t)\}$ with $(S_0, y_0) = (s, y_s)$, but with $\hat{e}_0 = \hat{e}$ and $e_0 = e$. By Lemma 2.1, $\hat{e}_t - e_t \xrightarrow{a.s.} 0$. Therefore, except on a null set of sample paths, it holds for all bounded Lipschitz continuous functions f on $\mathcal{S} \times \mathcal{M} \times \mathbb{R}^n$ that⁷

$$\left| \int f d\mu_{\hat{x},t} - \int f d\mu_{x,t} \right| = \left| \frac{1}{t} \sum_{k=0}^{t-1} f(S_k, y_k, \hat{e}_k) - \frac{1}{t} \sum_{k=0}^{t-1} f(S_k, y_k, e_k) \right| \rightarrow 0. \quad (2.14)$$

By the a.s. weak convergence of $\mu_{\hat{x},t}$ to $\mu_{\hat{x}}$ just proved, except on a null set, $\int f d\mu_{\hat{x},t} \rightarrow \int f d\mu_{\hat{x}}$ for all such functions f . Combining this with (2.14) yields that almost surely, $\int f d\mu_{x,t} \rightarrow \int f d\mu_{\hat{x}}$ for all such f . By [5, Theorem 11.3.3], this implies that almost surely, $\mu_{x,t} \rightarrow \mu_{\hat{x}}$ weakly.

Thus we have proved that for all initial conditions $x = (s, y_s, e)$, $e \in \mathbb{R}^n$, $\{\mu_{x,t}\}$ converges weakly, almost surely, to the same invariant probability measure $\mu_{\hat{x}}$. Denote $\mu = \mu_{\hat{x}}$. Let us now use a coupling argument to show that for any initial condition x , $\{\mu_{x,t}\}$ also converges to μ , \mathbf{P}_x -a.s.

⁷Here we are using the same (S_k, y_k) , $k \leq t$ in the occupation probability measures $\mu_{\hat{x},t}$ and $\mu_{x,t}$. This is valid because the e_t 's do not affect the evolution of $\{(S_t, y_t)\}$ and are functions of these states and the given initial e_0 . If we call the $\mu_{x,t}$ here $\tilde{\mu}_{x,t}$ instead and define $\mu_{x,t}$ using another independent copy of $\{(S_t, y_t)\}$, then since the two sequences of occupation probability measures will have the same probability distribution, $\{\mu_{x,t}\}$ will have the same weak convergence property as $\{\tilde{\mu}_{x,t}\}$.

Consider $\{(S_t, y_t, e_t)\}$ with an arbitrary initial condition $x = (\bar{s}, \bar{y}, \bar{e})$. Let $\tau = \min\{t \mid (S_t, y_t) = (s, y_s)\}$ (the pair (s, y_s) is as in the proof above). Note that $\tau < \infty$ a.s., because (s, y_s) is a recurrent state of $\{(S_t, y_t)\}$. Define $(\tilde{S}_k, \tilde{y}_k) = (S_{\tau+k}, y_{\tau+k})$, $\tilde{e}_k = e_{\tau+k}$ for $k \geq 0$.

By the strong Markov property (see e.g. [19, Theorem 3.3]), $\{(\tilde{S}_k, \tilde{y}_k)\}_{k \geq 0}$ has the same probability distribution as the Markov chain $\{(S_t, y_t)\}$ that starts from $(S_0, y_0) = (s, y_s)$. Therefore, by the preceding proof, $\mathbf{P}_{\bar{x}}$ -almost surely, for all bounded continuous functions f on $\mathcal{S} \times \mathcal{M} \times \mathbb{R}^n$,

$$\lim_{m \rightarrow \infty} \frac{1}{m} \sum_{k=0}^{m-1} f(\tilde{S}_k, \tilde{y}_k, \tilde{e}_k) = \int f d\mu. \quad (2.15)$$

Denote $a \wedge b = \min\{a, b\}$. Using (2.15) and the fact $\tau < \infty$ a.s., we have that $\mathbf{P}_{\bar{x}}$ -almost surely,

$$\begin{aligned} \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{k=0}^{t-1} f(S_k, y_k, e_k) &= \lim_{t \rightarrow \infty} \left(\frac{1}{t} \sum_{k=0}^{t \wedge (\tau-1)} f(S_k, y_k, e_k) + \frac{1}{t} \sum_{k=\tau}^t f(S_k, y_k, e_k) \right) \\ &= \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{k=0}^{t-\tau-1} f(S_{\tau+k}, y_{\tau+k}, e_{\tau+k}) \\ &= \lim_{m \rightarrow \infty} \frac{1}{m} \sum_{k=0}^{m-1} f(\tilde{S}_k, \tilde{y}_k, \tilde{e}_k) = \int f d\mu. \end{aligned}$$

This proves that $\{\mu_{x,t}\}$ converges weakly to μ almost surely, for each initial condition x .

It now follows that μ must be the unique invariant probability measure of $\{(S_t, y_t, e_t)\}$. To see this, suppose ζ is another invariant probability measure. For any bounded continuous function f , by stationarity, $\mathbb{E}_{\zeta}[\frac{1}{t} \sum_{k=0}^{t-1} f(S_k, y_k, e_k)] = \int f d\zeta$ for all $t \geq 1$. On the other hand, the preceding proof has established that for all initial conditions x , $\frac{1}{t} \sum_{k=0}^{t-1} f(S_k, y_k, e_k) = \int f d\mu_{x,t} \rightarrow \int f d\mu$, \mathbf{P}_x -a.s., which implies that if ζ is the initial distribution of (S_0, y_0, e_0) , then $\frac{1}{t} \sum_{k=0}^{t-1} f(S_k, y_k, e_k) \rightarrow \int f d\mu$, \mathbf{P}_{ζ} -a.s. We thus have

$$\begin{aligned} \int f d\zeta &= \mathbb{E}_{\zeta} \left[\frac{1}{t} \sum_{k=0}^{t-1} f(S_k, y_k, e_k) \right] = \lim_{t \rightarrow \infty} \mathbb{E}_{\zeta} \left[\frac{1}{t} \sum_{k=0}^{t-1} f(S_k, y_k, e_k) \right] \\ &= \mathbb{E}_{\zeta} \left[\lim_{t \rightarrow \infty} \frac{1}{t} \sum_{k=0}^{t-1} f(S_k, y_k, e_k) \right] = \int f d\mu, \end{aligned}$$

where the third equality follows from the bounded convergence theorem. This shows $\int f d\zeta = \int f d\mu$ for all bounded continuous functions f , and hence $\zeta = \mu$ by [5, Prop. 11.3.2], proving the uniqueness of the invariant probability measure.

The conclusions for the Markov chain $\{(S_t, S_{t+1}, y_t, e_t)\}$ follow from the same arguments given above, if we replace S_t with (S_t, S_{t+1}) and replace the set \mathcal{S} with the set of possible state transitions. (We could have proved the assertions for $\{(S_t, S_{t+1}, y_t, e_t)\}$ first and then deduced as their implications the assertions for $\{(S_t, y_t, e_t)\}$. We treated the latter first, as it makes the notation in the proof simpler.) \square

Remark 2.1 (About the proof). Theorem 2.1 is similar to [36, Theorem 3.2] for off-policy LSTD with constant λ (whose analysis also applies to state-dependent λ). Some of the techniques used to prove the two theorems are also similar. The main difference to [36] is that in the proof here we used a coupling argument to extend the weak convergence property of $\{\mu_{x,t}\}$ for a subset of initial conditions $x \in \{(s, y_s, e) \mid e \in \mathbb{R}^n\}$ to all initial conditions, whereas in [36] this step was proved using a result on the convergence-in-mean of LSTD iterates established first. The latter approach would not work here due to the dependence of λ_t on the history. Indeed, due to this dependence, the proof of the convergence-in-mean of LSTD given in [36] does not carry over to our case, even though that convergence does hold as a consequence of Theorem 2.1, in view of the boundedness of traces by construction. Compared with the proof of the ergodicity result in [36], the proof we gave here is more direct and therefore better.

Regarding possible alternative proofs of Theorem 2.1, let us also mention that if we prove first the uniqueness of the invariant probability measure, then, since $\{(S_t, y_t, e_t)\}_{t \geq 1}$ lie in a bounded set, the weak convergence of occupation probability measures will follow immediately from [16, Prop. 4.2].

However, because the evolution of the λ_t 's depends on both states and traces, it does not seem easy to us to prove directly the uniqueness part first. \square

Remark 2.2 (About the conditions on λ). Our proof of Theorem 2.1 relied on Lemma 2.1 and the two properties discussed preceding that lemma, namely, that $\{(S_t, y_t, e_t)\}$ is a weak Feller Markov chain and has at least one invariant probability measure. As long as these hold when we weaken or change the conditions on λ , the proof and the conclusions of the theorem will remain applicable.

We introduced Condition 2.3(ii) to bound the traces for algorithmic concerns. For the ergodicity of the state-trace process, Condition 2.3(ii) is unimportant—in fact, it can be removed from the conditions of Theorem 2.1. The reason is that we used this condition before Lemma 2.1 to quickly infer that $\{(S_t, y_t, e_t)\}$ has at least one invariant probability measure, but the latter is still true without it, in view of [17, Theorem 12.1.2(ii)] and the fact that $\{e_t\}$ is bounded in probability under Condition 2.1(i).⁸

Condition 2.3(i) is actually two conditions combined into one. The first is the continuity of $\lambda(y, e)e$ in e , which was used to ensure that the state-trace process is a weak Feller Markov chain. To be more general, instead of letting the evolutions of the traces and memory states be governed by the functions λ and g , one may consider letting them be governed by stochastic kernels. Then by placing a suitable continuity condition on the stochastic kernel λ , one can ensure that the state-trace process has the desired weak Feller Markov property.

The second condition packed into Condition 2.3(i) is that for each y , $\lambda(y, e)e$ is a Lipschitz continuous function of e with modulus 1. This condition is somewhat restrictive, and one may consider instead allowing the function to have Lipschitz modulus greater than 1. However, additional conditions are then needed to ensure that Lemma 2.1 holds. (If this lemma does not hold, then the state-trace process may not be ergodic and one will need a different approach than the one we took to characterize the sample path properties of the state-trace process.)

From an algorithmic perspective, if it is desirable to choose even larger λ_t 's or to have greater flexibility in choosing these λ -parameters, some of the generalizations just mentioned can be considered. For example, Condition 2.3(ii) can be replaced and stochastic kernels can be introduced to allow for occasionally large traces e_t , so that instead of having the traces bounded, one only make their variances bounded in a desired range. \square

3 Generalized Bellman Equations

In this section we continue the analysis started in Section 2.3. Our goal is to relate the linear equation in v , $\mathbb{E}_\zeta[e_0 \delta_0(v)] = 0$, which is the asymptotic limit of the linear equations (2.4) for LSTD as shown by Corollary 2.1, to a generalized Bellman equation for the target policy π . This will then allow us to interpret solutions of (2.4) as solutions of approximate versions of that generalized Bellman equation. To this end, we will first give in Section 3.1 a general description of randomized stopping times and associated generalized Bellman operators.

To simplify notation in subsequent derivations, we shall use the following shorthand notation: For $k \leq m$, denote $S_k^m = (S_k, S_{k+1}, \dots, S_m)$,

$$\rho_k^m = \prod_{i=k}^m \rho_i, \quad \lambda_k^m = \prod_{i=k}^m \lambda_i, \quad \gamma_k^m = \prod_{i=k}^m \gamma_i, \quad (3.1)$$

and by convention we treat $\rho_k^m = \lambda_k^m = \gamma_k^m = 1$ if $k > m$.

3.1 Randomized Stopping Times

Consider the Markov chain $\{S_t\}$ induced by the target policy π . Let Condition 2.1(i) hold. Recall that for the value function v_π , we have

$$v_\pi(s) = \mathbb{E}_s^\pi \left[\sum_{t=0}^{\infty} \gamma_1^t r_\pi(S_t) \right] \quad (\text{by definition}), \quad \text{and} \quad v_\pi(s) = r_\pi(s) + \mathbb{E}_s^\pi [\gamma_1 v_\pi(S_1)]$$

⁸The proof of this fact is straightforward and similar to the proof of [36, Lemma 3.1] or [37, Prop. A.1].

for each state s . The second equation is the standard one-step Bellman equation.

To write generalized Bellman equations for π , we shall make use of *randomized stopping times* for $\{S_t\}$, a notion that generalizes naturally stopping times for $\{S_t\}$ in that whether to stop at time t depends not only on the past states S_0^t but also on certain random outcomes. A simple example is to toss a coin at each time and stop as soon as the coin lands on heads, regardless of the history S_0^t . (The corresponding Bellman equation is the one associated with TD(λ) for a constant λ .) Of interest here is the general case where the stopping decision does depend on the entire history.

To define a randomized stopping time formally, first, the probability space of $\{S_t\}$ is enlarged to take into account whatever randomization scheme that is used to make the stopping decision. (The enlargement will be problem-dependent, as the next subsection will demonstrate.) Then, on the enlarged space, a randomized stopping time τ for $\{S_t\}$ is a stopping time⁹ relative to some increasing sequence of σ -algebras $\mathcal{F}_0 \subset \mathcal{F}_1 \subset \dots$, where the sequence $\{\mathcal{F}_t\}$ is such that

- (i) for all $t \geq 0$, $\mathcal{F}_t \supset \sigma(S_0^t)$ (the σ -algebra generated by S_0^t), and
- (ii) relative to $\{\mathcal{F}_t\}$, $\{S_t\}$ remains to be a Markov chain with transition probability P , i.e., for all $s \in \mathcal{S}$, $\text{Prob}(S_{t+1} = s \mid \mathcal{F}_t) = P_{S_t s}$.

See [19, Chap. 3.3] (in particular, see several equivalent definitions and Prop. 3.6 in p. 31-32).

The advantage of this abstract definition is that it allows us to write Bellman equations in general forms without worrying about the details of the enlarged space which are not important at this point. For notational simplicity, we shall still use \mathbb{E}^π to denote expectation for the enlarged probability space and write \mathbf{P}^π for the probability measure on that space, when there is no confusion.

If τ is a randomized stopping time for $\{S_t\}$, the strong Markov property [19, Theorem 3.3] allows us to express v_π in terms of $v_\pi(S_\tau)$ and the total discounted rewards R^τ prior to stopping:

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_s^\pi \left[\sum_{t=0}^{\tau-1} \gamma_1^t r_\pi(S_t) + \sum_{t=\tau}^{\infty} \gamma_1^t \cdot \gamma_{\tau+1}^t r_\pi(S_t) \right] \\ &= \mathbb{E}_s^\pi [R^\tau + \gamma_1^\tau v_\pi(S_\tau)], \end{aligned} \quad (3.2)$$

where $R^\tau = \sum_{t=0}^{\tau-1} \gamma_1^t r_\pi(S_t)$ for $\tau \in \{0, 1, 2, \dots\} \cup \{+\infty\}$.¹⁰ We can also write the Bellman equation (3.2) in terms of $\{S_t\}$ only, by taking expectation over τ :

$$v_\pi(s) = \mathbb{E}_s^\pi \left[\sum_{t=0}^{\infty} \left(q_t^+(S_0^t) \cdot \gamma_1^t r_\pi(S_t) + q_t(S_0^t) \cdot \gamma_1^t v_\pi(S_t) \right) \right], \quad (3.3)$$

$$\text{where } q_t^+(S_0^t) = \mathbf{P}^\pi(\tau > t \mid S_0^t), \quad q_t(S_0^t) = \mathbf{P}^\pi(\tau = t \mid S_0^t). \quad (3.4)$$

The right-hand side (r.h.s.) of (3.2) or (3.3) defines an associated generalized Bellman operator $T : \mathbb{R}^N \rightarrow \mathbb{R}^N$ that has several equivalent expressions; e.g.,

$$(Tv)(s) = \mathbb{E}_s^\pi [R^\tau + \gamma_1^\tau v(S_\tau)] = \mathbb{E}_s^\pi \left[\sum_{t=0}^{\infty} \left(q_t^+(S_0^t) \cdot \gamma_1^t r_\pi(S_t) + q_t(S_0^t) \cdot \gamma_1^t v(S_t) \right) \right], \quad s \in \mathcal{S}.$$

If $\tau \geq 1$ a.s., then just as in the case of the one-step Bellman operator, the value function v_π is the unique fixed point of T , i.e., the unique solution of $v = Tv$. In fact, this can be shown for slightly more general τ :

Theorem 3.1. *Let Condition 2.1(i) hold, and let the randomized stopping time τ be such that $\mathbf{P}^\pi(\tau \geq 1 \mid S_0 = s) > 0$ for all states $s \in \mathcal{S}$. Then v_π is the unique fixed point of the generalized Bellman operator T associated with τ , and T is a contraction w.r.t. a weighted sup-norm on \mathbb{R}^N .*

⁹A random time τ is a stopping time relative to a sequence $\{\mathcal{F}_t\}$ of increasing σ -algebras if the event $\{\tau = t\} \in \mathcal{F}_t$ for every t .

¹⁰In the case $\tau = 0$, $R^0 = 0$. In the case $\tau = \infty$, by Condition 2.1(i), $R^\infty = \sum_{t=0}^{\infty} \gamma_1^t r_\pi(S_t)$ is almost surely well-defined, while the second term $\gamma_1^\tau v_\pi(S_\tau)$ in (3.2) is 0 because $\gamma_1^\infty := \prod_{k=1}^{\infty} \gamma_k = 0$ a.s., under Condition 2.1(i).

We prove this theorem in Appendix A. The proof amounts to showing that if a state process evolves according to the substochastic matrix \tilde{P} involved in the affine operator T , then all the states in \mathcal{S} are transient (equivalently, the spectral radius of \tilde{P} is less than 1 and $I - \tilde{P}$ is invertible [22, Appendix A.4]). From this the conclusions of the theorem follow as a basic fact of nonnegative matrix theory [27, Theorem 1.1], and one specific choice of the weights of the sup-norm in the theorem is simply the expected time for the process to leave \mathcal{S} from each initial state (see e.g., the proof of [1, Prop. 2.2]).

3.2 Bellman Equation for the Proposed TD Learning Scheme

With the terminology of randomized stopping times, we are now ready to write down the generalized Bellman equation associated with the TD learning scheme proposed in Section 2.2. It corresponds to a particular randomized stopping time. We shall first describe this random time, from which a generalized Bellman equation follows as seen in the preceding subsection. That this is indeed the Bellman equation for our TD learning scheme will then be proved.

Consider the Markov chain $\{S_t\}$ under the target policy π . We define a randomized stopping time τ for $\{S_t\}$:

- Let $y_t, \lambda_t, e_t, t \geq 1$, evolve according to (2.5) and (2.2).
- Let the initial (S_0, y_0, e_0) be distributed according to ζ , the unique invariant probability measure in Theorem 2.1.
- At time $t \geq 1$, we stop the system with probability $1 - \lambda_t$ if it has not yet been stopped. Let τ be the time when the system stops ($\tau = \infty$ if the system never stops).

To make the dependence on the initial distribution ζ explicit, we write \mathbf{P}_ζ^π for the probability measure of this process.

Note that by definition λ_t and λ_1^t are functions of the initial (y_0, e_0) and states S_0^t . From how the random time τ is defined, we have for all $t \geq 1$,

$$\mathbf{P}_\zeta^\pi(\tau > t \mid S_0^t, y_0, e_0) = \lambda_1^t =: h_t^+(y_0, e_0, S_0^t), \quad (3.5)$$

$$\mathbf{P}_\zeta^\pi(\tau = t \mid S_0^t, y_0, e_0) = \lambda_1^{t-1}(1 - \lambda_t) =: h_t(y_0, e_0, S_0^t), \quad (3.6)$$

and hence

$$q_t^+(S_0^t) := \mathbf{P}_\zeta^\pi(\tau > t \mid S_0^t) = \int h_t^+(y, e, S_0^t) \zeta(d(y, e) \mid S_0), \quad (3.7)$$

$$q_t(S_0^t) := \mathbf{P}_\zeta^\pi(\tau = t \mid S_0^t) = \int h_t(y, e, S_0^t) \zeta(d(y, e) \mid S_0), \quad (3.8)$$

where $\zeta(d(y, e) \mid s)$ is the conditional distribution of (y_0, e_0) given $S_0 = s$, w.r.t. the initial distribution ζ . As before, we can write the generalized Bellman operator T associated with τ in several equivalent forms. Let \mathbb{E}_ζ^π denote expectation under \mathbf{P}_ζ^π . Based on (3.2) and (3.5)-(3.6), it is easy to derive that¹¹ for all $v : \mathcal{S} \rightarrow \mathbb{R}, s \in \mathcal{S}$,

$$(Tv)(s) = \mathbb{E}_\zeta^\pi \left[\sum_{t=0}^{\infty} \lambda_1^t \gamma_1^t r_\pi(S_t) + \sum_{t=1}^{\infty} \lambda_1^{t-1} (1 - \lambda_t) \gamma_1^t v(S_t) \mid S_0 = s \right]. \quad (3.9)$$

Alternatively, by integrating over (y_0, e_0) and using (3.7)-(3.8), we can write similarly to (3.3) that

$$(Tv)(s) = \mathbb{E}_\zeta^\pi \left[\sum_{t=0}^{\infty} \left(q_t^+(S_0^t) \cdot \gamma_1^t r_\pi(S_t) + q_t(S_0^t) \cdot \gamma_1^t v(S_t) \right) \mid S_0 = s \right], \quad (3.10)$$

for all $v : \mathcal{S} \rightarrow \mathbb{R}, s \in \mathcal{S}$, where in the case $t = 0$, $q_0^+(S_0) = 1$ and $q_0(S_0) = 0$ since $\tau > 0$ by construction.

¹¹Rewrite (3.2) as $v_\pi(s) = \mathbb{E}_s^\pi \left[\sum_{t=0}^{\infty} \mathbb{1}(\tau > t) \gamma_1^t r_\pi(S_t) + \sum_{t=0}^{\infty} \mathbb{1}(\tau = t) \gamma_1^t v_\pi(S_t) \right]$ and for the t th terms in the r.h.s., take expectation over τ conditioned on (S_0^t, y_0, e_0) .

Remark 3.1. Comparing the two expressions of T , we remark that the expression (3.9) reflects the role of the λ_t 's in determining the stopping time, whereas the expression (3.10), which has eliminated the auxiliary variables y_t and e_t , shows more clearly the dependence of the stopping time on the entire history S_0^t . It can also be seen, from the initial distribution ζ , the dependence of λ_t on the traces and the dependence of the traces on the function ρ , that both the behavior policy and the choice of the feature representation assert a significant role in determining the Bellman operator T for the target policy. This is in contrast with off-policy TD learning that uses a constant λ , where the behavior policy and the approximation subspace affect only how one approximates the Bellman equation underlying TD, not the Bellman equation itself, which is solely determined by λ . Furthermore, note that as the invariant distribution of the state-trace process, ζ is associated with the dynamic behavior of the states *and traces* under the behavior policy, and generally, there is no explicit expression of ζ in terms of P^o and the parameters in the λ function. As a result, in general we cannot express the operator T in terms of these parameters in the learning scheme. This is different from the case of $\text{TD}(\lambda)$ where λ is a function of the present state only. \square

We now proceed to show how the Bellman equation $v = Tv$ given above relates to the off-policy TD learning scheme in Section 2.2. Some notation is needed. Denote by $\zeta_{\mathcal{S}}$ the invariant probability measure of the Markov chain $\{S_t\}$ induced by the behavior policy; note that it coincides with the marginal of ζ on \mathcal{S} . For two functions v_1, v_2 on \mathcal{S} , we write $v_1 \perp_{\zeta_{\mathcal{S}}} v_2$ if $\sum_{s \in \mathcal{S}} \zeta_{\mathcal{S}}(s) v_1(s) v_2(s) = 0$. If \mathcal{L} is a linear subspace of functions on \mathcal{S} and $v \perp_{\zeta_{\mathcal{S}}} v'$ for all $v' \in \mathcal{L}$, we write $v \perp_{\zeta_{\mathcal{S}}} \mathcal{L}$. Recall that ϕ is a function that maps each state s to an n -dimensional feature vector. Denote by \mathcal{L}_{ϕ} the subspace spanned by the n component functions of ϕ , which is the space of approximate value functions for our TD learning scheme. Recall also that \mathbb{E}_{ζ} denotes expectation w.r.t. the *stationary* state-trace process $\{(S_t, y_t, e_t)\}$ under the behavior policy (cf. Theorem 2.1).

Theorem 3.2. *Let Conditions 2.1-2.3 hold. Then as a linear equation in v , $\mathbb{E}_{\zeta}[e_0 \delta_0(v)] = 0$ is equivalently $Tv - v \perp_{\zeta_{\mathcal{S}}} \mathcal{L}_{\phi}$, where T is the generalized Bellman operator for π given in (3.9) or (3.10).*

Remark 3.2 (On LSTD). Note that

$$Tv - v \perp_{\zeta_{\mathcal{S}}} \mathcal{L}_{\phi}, \quad v \in \mathcal{L}_{\phi}$$

is a projected version of the generalized Bellman equation $Tv - v = 0$ (projecting the left-hand side onto the approximation subspace \mathcal{L}_{ϕ} w.r.t. the $\zeta_{\mathcal{S}}$ -weighted Euclidean norm). Theorem 3.2 and Corollary 2.1 together show that this is what LSTD solves in the limit. If this projected Bellman equation admits a unique solution \bar{v} , then \bar{v} can be viewed as an oblique projection of v_{π} [26] and the approximation error $\bar{v} - v_{\pi}$ can be characterized as in [40]. The details of these are given in Appendix B. \square

Let us give a corollary to Theorem 3.2 before proving it. The corollary concerns a composite scheme of setting λ , which is slightly more general than what Section 2.2 described and can be useful in practice for variance control. Let us describe the scheme first, deferring the discussion about the motivation temporarily.

Partition the state space into m nonempty disjoint sets: $\mathcal{S} = \cup_{i=1}^m \mathcal{S}_i$. Associate each set \mathcal{S}_i with a possibly different scheme of setting λ that is of the type described in Section 2.2, and denote its memory states by $y_t^{(i)}$ and λ -function by $\lambda^{(i)}(\cdot, \cdot)$. Keep m trace vectors $e_t^{(1)}, \dots, e_t^{(m)}$, one for each set, and update them according to

$$e_t^{(i)} = \lambda_t^{(i)} \gamma_t \rho_{t-1} e_{t-1}^{(i)} + \phi(S_t) \mathbb{1}(S_t \in \mathcal{S}_i), \quad 1 \leq i \leq m, \quad (3.11)$$

where $\lambda_t^{(i)} = \lambda^{(i)}(y_t^{(i)}, e_{t-1}^{(i)})$. We then have m ergodic state-trace processes that share the same state variables, $\{(S_t, y_t^{(i)}, e_t^{(i)})\}$, $i = 1, 2, \dots, m$. Each process has a unique invariant probability

measure $\zeta^{(i)}$ (Theorem 2.1) and an associated randomized stopping time $\tau^{(i)}$ and generalized Bellman operator $T^{(i)}$, as discussed in this subsection. Define now an operator T by concatenating the component mappings of $T^{(i)}$ for \mathcal{S}_i as follows: for all $v \in \mathfrak{R}^N$ and $s \in \mathcal{S}$,

$$(Tv)(s) := (T^{(i)}v)(s) \quad \text{if } s \in \mathcal{S}_i. \quad (3.12)$$

Consider an LSTD algorithm that defines the trace e_t to be the sum of the m trace vectors,

$$e_t = \sum_{i=1}^m e_t^{(i)}, \quad (3.13)$$

and uses the traces to form the linear equation as before,

$$\frac{1}{t} \sum_{k=0}^{t-1} e_k \delta_k(v) = 0, \quad v = \Phi\theta.$$

Note that $\frac{1}{t} \sum_{k=0}^{t-1} e_k \delta_k(v) = 0$ is equivalent to $\sum_{i=1}^m \frac{1}{t} \sum_{k=0}^{t-1} e_k^{(i)} \delta_k(v) = 0$, and as a linear equation in v , it tends asymptotically (as $t \rightarrow \infty$) to the linear equation $\sum_{i=1}^m \mathbb{E}_{\zeta^{(i)}} [e_0^{(i)} \delta_0(v)] = 0$ (Corollary 2.1).

Corollary 3.1. *Let Condition 2.1 hold. Consider the composite scheme of setting λ discussed above, and let Conditions 2.2-2.3 hold for each of the m schemes involved. Let LSTD calculate traces according to (3.11) and (3.13). Then the limiting linear equation (in v) associated with LSTD, $\sum_{i=1}^m \mathbb{E}_{\zeta^{(i)}} [e_0^{(i)} \delta_0(v)] = 0$, is equivalently $Tv - v \perp_{\zeta_S} \mathcal{L}_\phi$, where T is the generalized Bellman operator for π given by (3.12) and has the same fixed point and contraction properties as stated in Theorem 3.1.*

This corollary will be proved in the next subsection. The use of composite schemes will be demonstrated by experiments in Section 4.2.2. Here let us explain informally our motivation for such schemes.

Remark 3.3 (About composite schemes of setting λ). Our motivation is revealed by the equation (3.12). Typically each $T^{(i)}$ is designed to be simple to implement in TD learning. For example, if we ignore for now the bounding of traces introduced in Section 2.2 and just consider TD(λ) with constant λ , $T^{(i)}$ can be the Bellman operator $T^{(\lambda)}$ for TD(λ) with some constant λ . A simple, extreme example is to partition the state space into two sets, and associate one with $T^{(\lambda)}$, $\lambda = 1$, and the other with $T^{(\lambda)}$, $\lambda = 0$. Using the combination (3.12) of the two operators in TD then means that for the first set of states, we want to estimate their values by using the information about the total rewards received when starting from those states, whereas for the second set of states, we only use the information about their one-stage rewards and how these states relate to their “neighboring” states in the transition graph. While this way of using different kinds of information for different states is natural and useful for TD-based policy evaluation, it cannot be realized by keeping a single trace vector and letting λ_t evolve with states or histories. Indeed, in that case $\lambda_t = 1$ does not mean that the information about the total rewards from state S_t will be used, because the subsequent $\lambda_{t+1}, \lambda_{t+2}, \dots$ could be small or even zero.

In the context of the more complex scheme of setting λ discussed in this paper, our motivation and reasons for considering composite schemes are the same. Each $T^{(i)}$ can be designed to be simple to implement, such as in the simple scaling example in Section 2.2. The parameters in the i th scheme can be chosen so that they encourage the use of large λ_t ’s throughout time or dictate the use of only small λ_t ’s. By combining component mappings of $T^{(i)}$ through (3.12), composite schemes allow us to use different kinds of information for different states in estimating the value function.

Finally, we mention that for off-policy LSTD(λ) with constant λ , composite schemes have been considered and analyzed in [36, Proposition 4.5, Section 4.3]. Our Corollary 3.1 above can be viewed as an extension of that result. \square

3.3 Proofs of Theorem 3.2 and Corollary 3.1

We divide the proof of Theorem 3.2 into three steps. The first step deals with an expression for the trace vector, given in the following lemma. It is more subtle than the other two steps, which involve mostly calculations.

We start by extending the stationary state-trace process $\{(S_t, y_t, e_t)\}_{t \geq 0}$ to $t = -1, -2, \dots$, and work with a double-ended stationary process $\{(S_t, y_t, e_t)\}_{-\infty < t < \infty}$ (by Kolmogorov's theorem [5, Theorem 12.1.2], such a process exists). Note that as before this is a Markov chain whose transition probability is defined by the behavior policy together with the update rules (2.2) and (2.5) for e_t , y_t and λ_t , and the marginal distribution of each (S_t, y_t, e_t) is ζ . We keep using the notation \mathbf{P}_ζ and \mathbb{E}_ζ for this double-ended stationary Markov chain.

Recall the shorthand notation (3.1) introduced at the beginning of Section 3: For $k \leq m$, $\rho_k^m = \prod_{i=k}^m \rho_i$, $\lambda_k^m = \prod_{i=k}^m \lambda_i$, $\gamma_k^m = \prod_{i=k}^m \gamma_i$, and in addition, $\lambda_1^0 = \gamma_1^0 = \rho_0^{-1} = 1$ by convention.

Lemma 3.1. *\mathbf{P}_ζ -almost surely, $\sum_{t=1}^\infty \lambda_{1-t}^0 \gamma_{1-t}^0 \rho_{-t}^{-1} \phi(S_{-t})$ is well-defined and finite, and*

$$e_0 = \phi(S_0) + \sum_{t=1}^\infty \lambda_{1-t}^0 \gamma_{1-t}^0 \rho_{-t}^{-1} \phi(S_{-t}). \quad (3.14)$$

Proof. First, we show $\mathbb{E}_\zeta[\sum_{t=1}^\infty \gamma_{1-t}^0 \rho_{-t}^{-1}] < \infty$. Indeed,

$$\mathbb{E}_\zeta[\sum_{t=1}^\infty \gamma_{1-t}^0 \rho_{-t}^{-1}] = \sum_{t=1}^\infty \mathbb{E}_\zeta[\gamma_{1-t}^0 \rho_{-t}^{-1}] = \sum_{t=1}^\infty \zeta^\top (P\Gamma)^t \mathbf{1} < \infty,$$

where the first equality follows from the monotone convergence theorem, the second equality from Condition 2.1(ii) and a direct calculation, and the last inequality follows from Condition 2.1(i) (since $(I - P\Gamma)^{-1} = \sum_{t=0}^\infty (P\Gamma)^t$). This implies $\sum_{t=1}^\infty \gamma_{1-t}^0 \rho_{-t}^{-1} < \infty$, \mathbf{P}_ζ -a.s., so $\gamma_{1-t}^0 \rho_{-t}^{-1} \rightarrow 0$ as $t \rightarrow \infty$, \mathbf{P}_ζ -a.s. Since $\lambda_{1-t}^0 \leq 1$ for all t , it also implies that

$$\mathbb{E}_\zeta[\sum_{t=1}^\infty \lambda_{1-t}^0 \gamma_{1-t}^0 \rho_{-t}^{-1} \|\phi(S_{-t})\|] \leq \max_{s \in \mathcal{S}} \|\phi(s)\| \cdot \mathbb{E}_\zeta[\sum_{t=1}^\infty \gamma_{1-t}^0 \rho_{-t}^{-1}] < \infty.$$

It follows from a theorem on integration [24, Theorem 1.38, p. 28-29] that \mathbf{P}_ζ -almost surely, the infinite series $\sum_{t=1}^\infty \lambda_{1-t}^0 \gamma_{1-t}^0 \rho_{-t}^{-1} \phi(S_{-t})$ converges to a finite limit.

We now prove the expression for e_0 . By unfolding the iteration (2.2) for e_t backwards in time, we have for all $m \geq 1$,

$$e_0 = \phi(S_0) + \sum_{t=1}^{m-1} \lambda_{1-t}^0 \gamma_{1-t}^0 \rho_{-t}^{-1} \phi(S_{-t}) + \lambda_{1-m}^0 \gamma_{1-m}^0 \rho_{-m}^{-1} e_{-m}. \quad (3.15)$$

Let $m \rightarrow \infty$ in the r.h.s. of (3.15). For the last term, the trace e_{-m} lies in a bounded set by Condition 2.3(ii), $\lambda_{1-m}^0 \leq 1$, and as we just showed, $\gamma_{1-m}^0 \rho_{-m}^{-1} \rightarrow 0$, \mathbf{P}_ζ -a.s. So the last term converges to zero \mathbf{P}_ζ -a.s. Also as we just showed, the second term converges \mathbf{P}_ζ -almost surely to $\sum_{t=1}^\infty \lambda_{1-t}^0 \gamma_{1-t}^0 \rho_{-t}^{-1} \phi(S_{-t})$. The expression (3.14) for e_0 then follows. \square

Proof of Theorem 3.2. Treating $\lambda_1^0 = \gamma_1^0 = \rho_0^{-1} = 1$, we write the expression of e_0 given in Lemma 3.1 as $e_0 = \sum_{t=0}^\infty \lambda_{1-t}^0 \gamma_{1-t}^0 \rho_{-t}^{-1} \phi(S_{-t})$, \mathbf{P}_ζ -a.s. We use this expression to calculate first $\mathbb{E}_\zeta[e_0 \cdot \rho_0 f(S_0^1)]$ for an arbitrary bounded measurable function f on $\mathcal{S} \times \mathcal{S}$. We have

$$\begin{aligned} \mathbb{E}_\zeta[e_0 \cdot \rho_0 f(S_0^1)] &= \sum_{t=0}^\infty \mathbb{E}_\zeta[\lambda_{1-t}^0 \gamma_{1-t}^0 \rho_{-t}^{-1} \phi(S_{-t}) \cdot \rho_0 f(S_0^1)] \\ &= \sum_{t=0}^\infty \mathbb{E}_\zeta[\lambda_1^t \gamma_1^t \rho_0^{t-1} \phi(S_0) \cdot \rho_t f(S_t^{t+1})] \\ &= \sum_{t=0}^\infty \mathbb{E}_\zeta[\phi(S_0) \cdot \mathbb{E}_\zeta[\lambda_1^t \gamma_1^t \rho_0^t f(S_t^{t+1}) \mid S_0]] \end{aligned} \quad (3.16)$$

where we used the stationarity of the double-ended state-trace process to derive the second equality, and we changed the order of expectation and summation in the first equality. The latter change is

justified by the dominated convergence theorem, and so are similar interchanges of expectation and summation that will appear in the rest of this proof.

To proceed with the calculation, we relate the expectations in the summation in (3.16) to expectations w.r.t. the process with probability measure \mathbf{P}_ζ^π introduced in Section 3.2 (which we recall is induced by the target policy π and involves the random stopping time τ). Let $\tilde{\mathbb{E}}_\zeta^\pi$ denote expectation w.r.t. the marginal of \mathbf{P}_ζ^π on the space of $\{(S_t, y_t, e_t)\}_{t \geq 0}$. From the change of measure performed through ρ_0^t , we have

$$\mathbb{E}_\zeta [\lambda_1^t \gamma_1^t \rho_0^t f(S_t^{t+1}) \mid S_0, y_0, e_0] = \tilde{\mathbb{E}}_\zeta^\pi [\lambda_1^t \gamma_1^t f(S_t^{t+1}) \mid S_0, y_0, e_0], \quad t \geq 0. \quad (3.17)$$

Combining this with (3.16) and using the fact that ζ is the marginal distribution of (S_0, y_0, e_0) in both processes, we obtain

$$\begin{aligned} \mathbb{E}_\zeta [e_0 \cdot \rho_0 f(S_0^1)] &= \sum_{t=0}^{\infty} \tilde{\mathbb{E}}_\zeta^\pi \left[\phi(S_0) \cdot \tilde{\mathbb{E}}_\zeta^\pi [\lambda_1^t \gamma_1^t f(S_t^{t+1}) \mid S_0] \right] \\ &= \tilde{\mathbb{E}}_\zeta^\pi \left[\phi(S_0) \cdot \sum_{t=0}^{\infty} \tilde{\mathbb{E}}_\zeta^\pi [\lambda_1^t \gamma_1^t f(S_t^{t+1}) \mid S_0] \right]. \end{aligned} \quad (3.18)$$

We now use (3.18) to calculate $\mathbb{E}_\zeta [e_0 \delta_0(v)]$ for a given function v . Recall from (2.3) that $\delta_0(v) = \rho_0 \cdot (r(S_0^1) + \gamma_1 v(S_1) - v(S_0))$, so we let $f(S_t^{t+1}) = r(S_t^{t+1}) + \gamma_{t+1} v(S_{t+1}) - v(S_t)$ in (3.18). In particular, since $\mathbb{E}_\zeta^\pi [r(S_t^{t+1}) \mid S_0^t] = r_\pi(S_t)$,

$$\sum_{t=0}^{\infty} \tilde{\mathbb{E}}_\zeta^\pi [\lambda_1^t \gamma_1^t r(S_t^{t+1}) \mid S_0] = \sum_{t=0}^{\infty} \tilde{\mathbb{E}}_\zeta^\pi [\lambda_1^t \gamma_1^t r_\pi(S_t) \mid S_0].$$

By rearranging terms, $\sum_{t=0}^{\infty} \tilde{\mathbb{E}}_\zeta^\pi [\lambda_1^t \gamma_1^t \cdot (\gamma_{t+1} v(S_{t+1}) - v(S_t)) \mid S_0]$ equals

$$-v(S_0) + \sum_{t=1}^{\infty} \tilde{\mathbb{E}}_\zeta^\pi [\lambda_1^{t-1} (1 - \lambda_t) \cdot \gamma_1^t v(S_t) \mid S_0].$$

Putting these together and using the expression of T in (3.9), we obtain

$$\begin{aligned} \mathbb{E}_\zeta [e_0 \delta_0(v) \mid S_0] &= \phi(S_0) \cdot \left\{ -v(S_0) + r_\pi(S_0) + \tilde{\mathbb{E}}_\zeta^\pi \left[\sum_{t=1}^{\infty} \left(\lambda_1^t \cdot \gamma_1^t r_\pi(S_t) + \lambda_1^{t-1} (1 - \lambda_t) \cdot \gamma_1^t v(S_t) \right) \mid S_0 \right] \right\} \\ &= \phi(S_0) \cdot \{ -v(S_0) + (Tv)(S_0) \}. \end{aligned}$$

Therefore

$$\mathbb{E}_\zeta [e_0 \delta_0(v)] = \sum_{s \in \mathcal{S}} \zeta_{\mathcal{S}}(s) \phi(s) \cdot (Tv - v)(s), \quad (3.19)$$

and this shows that $\mathbb{E}_\zeta [e_0 \delta_0(v)] = 0$ is equivalent to $Tv - v \perp_{\zeta_{\mathcal{S}}} \mathcal{L}_\phi$. \square

We now prove Corollary 3.1.

Proof of Corollary 3.1. We first apply Theorem 3.2 to each state-trace process $\{(S_t, y_t^{(i)}, e_t^{(i)})\}$ for $i = 1, 2, \dots, m$. Specifically, by (3.19) and the definition (3.11) of $e_t^{(i)}$,

$$\mathbb{E}_{\zeta^{(i)}} [e_0^{(i)} \delta_0(v)] = \sum_{s \in \mathcal{S}} \zeta_{\mathcal{S}}(s) \cdot \phi(s) \mathbb{1}(s \in \mathcal{S}_i) \cdot (T^{(i)}v - v)(s).$$

Hence

$$\begin{aligned} \sum_{i=1}^m \mathbb{E}_{\zeta^{(i)}} [e_0^{(i)} \delta_0(v)] &= \sum_{s \in \mathcal{S}} \zeta_{\mathcal{S}}(s) \phi(s) \cdot \left[\sum_{i=1}^m \mathbb{1}(s \in \mathcal{S}_i) \cdot (T^{(i)}v - v)(s) \right] \\ &= \sum_{s \in \mathcal{S}} \zeta_{\mathcal{S}}(s) \phi(s) \cdot (Tv - v)(s), \end{aligned}$$

where the last equality follows from the definition (3.12) of the operator T . This shows that the linear equation in v , $\sum_{i=1}^m \mathbb{E}_{\zeta^{(i)}} [e_0^{(i)} \delta_0(v)] = 0$, is equivalently $Tv - v \perp_{\zeta_{\mathcal{S}}} \mathcal{L}_\phi$.

We now prove that T has v_π as its unique fixed point and is a contraction with respect to a weighted sup-norm—in other words, Theorem 3.1 applies to T . For this, it suffices to show that T satisfies the conditions of Theorem 3.1, namely, T is a generalized Bellman operator associated with a randomized stopping time τ that satisfies $\mathbf{P}^\pi(\tau \geq 1 \mid S_0 = s) > 0$ for all states $s \in \mathcal{S}$. We can define such a random time τ from the randomized stopping times $\tau^{(i)}$ associated with the Bellman operators $T^{(i)}$, which, by their construction, all satisfy the conditions of Theorem 3.1. Specifically, by enlarging the probability space if necessary, we can regard $\tau^{(i)}$, $i = 1, 2, \dots, m$, as being defined on the same probability space.¹² We then let $\tau = \tau^{(i)}$ if $S_0 \in \mathcal{S}_i$. With this definition, we have $\mathbf{P}^\pi(\tau \geq 1 \mid S_0 = s) > 0$ for all states $s \in \mathcal{S}$ (since $\tau^{(i)} \geq 1$ a.s. for all i), and for each set \mathcal{S}_i , the corresponding component mappings of the generalized Bellman operator associated with τ coincide with those of $T^{(i)}$. In other words, T is the generalized Bellman operator associated with τ . \square

Remark 3.4. This completes our analysis of the LSTD algorithm for the proposed TD learning scheme. Note that the preceding results also prepare the ground for analyzing gradient-based algorithms similar to [12, 13] in a future work. Specifically, like LSTD, these algorithms would aim to solve the same projected generalized Bellman equation as characterized by Theorem 3.2 (cf. Remark 3.2). Their average dynamics, which is important for analyzing their convergence using the mean ODE approach from stochastic approximation theory [10], can be studied based on the ergodicity result of Theorem 2.1, in essentially the same way as we did in Section 2.3 for the LSTD algorithm. \square

4 Numerical Study

In this section we first use a toy problem to illustrate the behavior of traces calculated by off-policy LSTD(λ) for constant λ and for λ that evolves according to a simple special case of our proposed scheme. We then compare the behavior of LSTD for various choices of λ , on the toy problem and on the Mountain Car problem.

4.1 Behavior of Traces

The toy problem we use in this study has 21 states, arranged as shown in Figure 1 (left). One state is located at the centre, and the rest of the states split evenly into four groups, indicated by the four loops in the figure. The topology of the transition graph is the same for the target and behavior policies. We have drawn the transition graph only for the northeast group in Figure 1 (left); the states in each of the other three groups are arranged in the same manner and have the same transition structure. Given this symmetry, to specify the transition matrices P_π and P_{π° , it suffices to specify their submatrices for the central state and one of the groups. If we label the central state as state 1 and the states in the northeast group clockwise as states 2-6, the submatrices of P_π , P_{π° for these states are given, respectively, by

$$\pi : \begin{pmatrix} 0 & 0.25 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0.8 & 0 & 0 \\ 0 & 0 & 0.2 & 0 & 0.8 & 0 \\ 0 & 0 & 0 & 0.2 & 0 & 0.8 \\ 0.8 & 0 & 0 & 0 & 0.2 & 0 \end{pmatrix}, \quad \pi^\circ : \begin{pmatrix} 0 & 0.25 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 & 0 & 0.5 \\ 0.5 & 0 & 0 & 0 & 0.5 & 0 \end{pmatrix}.$$

Intuitively speaking, from the central state, the system enters one group of states by moving diagonally in one of the four directions with equal probability, and after spending some time in that

¹²That we can do so is clear from the definition of each $\tau^{(i)}$ as described at the beginning of Section 3.2 and from the fact that for each i , the initial distribution $\zeta^{(i)}$ on $(S_0, y_0^{(i)}, e_0^{(i)})$ has the same marginal on \mathcal{S} , which is $\zeta_\mathcal{S}$.

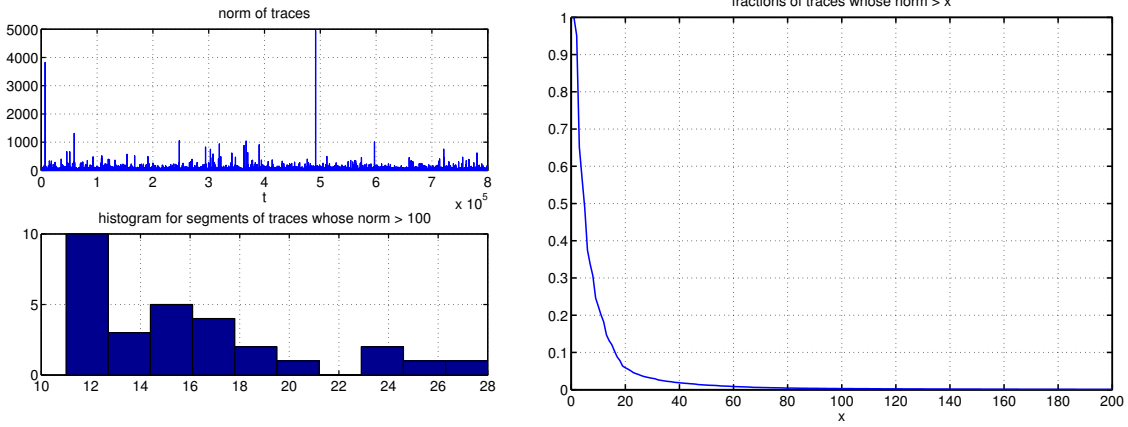


Figure 2: Statistics of traces for TD(1) in a toy problem (see the text for details).

histogram concerns the excursions of the trajectory $\{e_t\}$ outside of the ball $\{x \in \mathbb{R}^{n+1} \mid \|x\| \leq 100\}$. The horizontal axis indicates the lengths of the excursions (where the length is the number of iterations an excursion contains), and the vertical axis indicates how many excursions of length x occurred during the 8×10^5 iterations of the experimental run. We plotted the histogram for lengths $x > 10$. It can be seen that one can have large traces during many consecutive iterations. Such behavior, although tolerable by LSTD, is especially detrimental to TD algorithms and can disrupt their learning. This is our main motivation for suggesting the use of λ -parameters to bound the traces directly.

4.1.2 Traces with Evolving λ

We now illustrate the behavior of traces and LSTD for λ that evolves according to our proposed scheme. Specifically, for this demonstration, we will use the simple scaling example given by (2.6)-(2.7) in Example 2.1, with all the thresholds $C_{ss'}$ being the same constant C . We first simulate the state-trace process to illustrate the ergodicity of this process stated by Theorem 2.1, and we then study the performance of LSTD for different values of C . The results of these two experiments are shown in Figure 3 and Figures 4-5, respectively, and the details are as follows.

According to the ergodicity result of Theorem 2.1, no matter from which initial state and trace pair (S_0, e_0) we generate a trajectory $\{(S_t, e_t)\}_{0 \leq t \leq \bar{t}}$ according to the behavior policy, the empirical distribution of state and trace pairs in this trajectory should converge, as $\bar{t} \rightarrow \infty$, to the same distribution on $\mathcal{S} \times \mathbb{R}^n$, which is the marginal of the invariant probability measure ζ on that space. Since \mathcal{S} is discrete, we can verify this fact by examining the empirical conditional distribution of the trace given the state. In other words, for each state s , we examine the empirical distribution of the trace for the sub-trajectory (S_{t_k}, e_{t_k}) , $k = 1, 2, \dots$, where $S_{t_k} = s$ and it is the k th visit to state s by the trajectory. We check if this empirical distribution converges to the same one as we increase the length \bar{t} of the trajectory and as we vary the initial condition (S_0, e_0) .

To give a sense of what these limiting distributions over the trace space look like, we set the parameter $C = 50$ and generated a long trajectory with 8×10^5 iterations. In the top row of Figure 3, we plotted three normalized histograms of the first trace component for the sub-trajectories associated with three states of the toy problem, respectively: the central state (left), the middle state of the northeast group (middle), and the first state of the southeast group (right).

To check whether the empirical conditional distributions on the trace space converge along the sub-trajectory $\{(S_{t_k}, e_{t_k})\}$ for a given state, we compare the characteristic functions f_k of these distributions with the characteristic function f of the empirical conditional distribution obtained at the end of the sub-trajectory during the experimental run. In particular, we evaluate all these

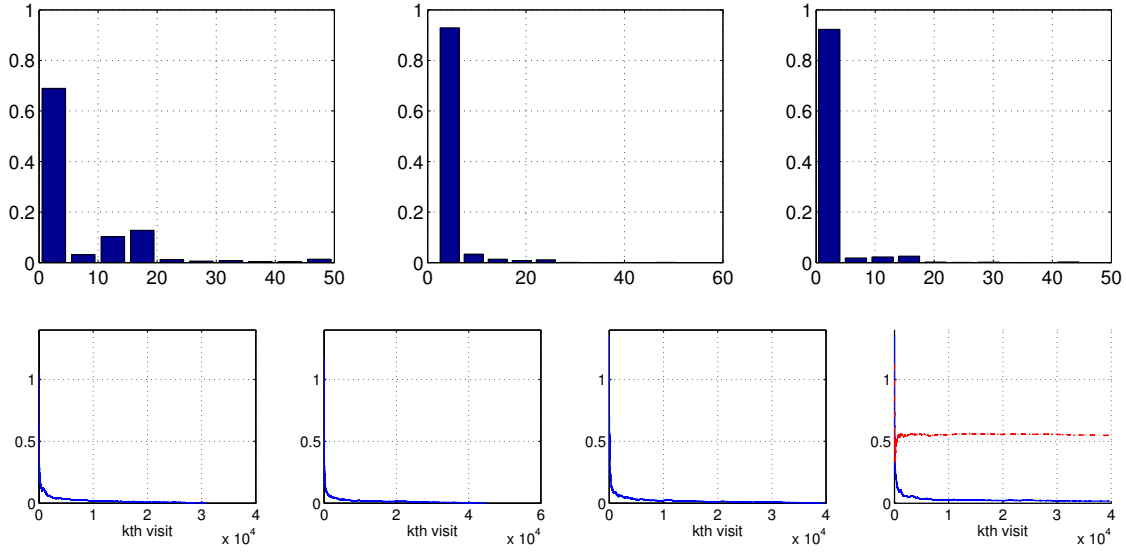


Figure 3: Convergence of empirical conditional distributions on the trace space.

(complex-valued) characteristic functions at 500 points, which are chosen randomly according to the normal distribution on \mathbb{R}^5 with mean 0 and covariance $200^2 I$. We take the maximal difference between f_k and f at these 500 points as an indicator of the deviation between the two corresponding distributions.¹³ In the bottom row of Figure 3, the first three plots show the difference curves obtained in the way just described, for three different states, respectively. These three states are the same ones mentioned earlier in the description of the top row of Figure 3. The horizontal axis of these plots indicates k , the number of visits to the corresponding state. As can be seen, the difference curves all tend to zero as k increases, which is consistent with the predicted convergence of the empirical conditional distributions on the trace space.

So far we compared the empirical distributions along the same trajectory. Next we compare them against the one obtained at the end of another trajectory that starts from a different initial condition. The difference curve for one state (the first state in the southeast group) is plotted in the last graph in the bottom row of Figure 3, and it is the lower curve in that graph. As can be seen, the curve tends to zero, suggesting that the limiting distribution of these empirical distributions does not change if we vary the initial condition, which is consistent with Theorem 2.1.

For comparison, we also plotted the difference curve when these same empirical conditional distributions are compared against the empirical conditional distribution obtained from the same trajectory but for a different state (specifically, the middle state of the northeast group). This is the upper curve in the last graph in the bottom row of Figure 3. It clearly indicates that for the two states, the associated limiting conditional distributions on the trace space are different. It also shows that the characteristic function approach we adopted in this experiment can effectively distinguish between two different distributions (cf. Footnote 13).

4.2 LSTD with Evolving λ

4.2.1 A Toy Problem

Let us first continue with the toy problem of the previous subsection and show how the LSTD algorithm performs in this problem as we vary the parameter C in the λ function for bounding the

¹³Recall that a tight sequence $\{p_k\}$ of probability distributions on \mathbb{R}^m converges to a probability distribution p if and only if the characteristic functions of p_k converge pointwise to the characteristic function of p [5, Lemma 9.5.5].

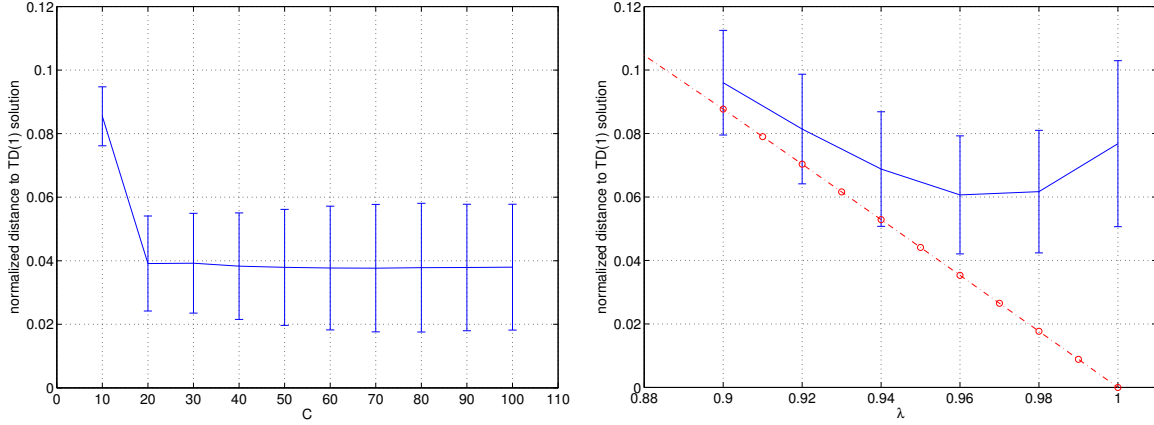


Figure 4: Compare LSTD solutions with evolving λ (left) and with constant λ (right).

traces. We ran LSTD for $C = 10, 20, \dots, 100$, using the same trajectory, for 3×10^5 iterations, and we computed the (Euclidean) distance of these LSTD solutions to the asymptotic TD(1) solution (in the space of the θ -parameters), normalized by the norm of the latter. We then repeat this calculation 10 times, each time with an independently generated trajectory. Plotted in Figure 4 (left) against the values of C are the means and standard deviations of the normalized distances of LSTD solutions thus obtained.

For comparison, we did the same for LSTD with a large constant λ : $\lambda = 0.9, 0.92, \dots, 1$. Figure 4 (right) shows the result, where the dash-dot curve indicates the normalized distance of the asymptotic TD(λ) solution—the solution that LSTD(λ) would obtain in the limit. It can be seen that the convergence of LSTD with a large constant λ is slow and requires more iterations than the 3×10^5 iterations performed. In comparison, LSTD with evolving λ works effectively for $C \geq 20$, and the approximation quality it achieved with such C is comparable to that of asymptotic TD(λ) solutions for a large constant λ around 0.96.

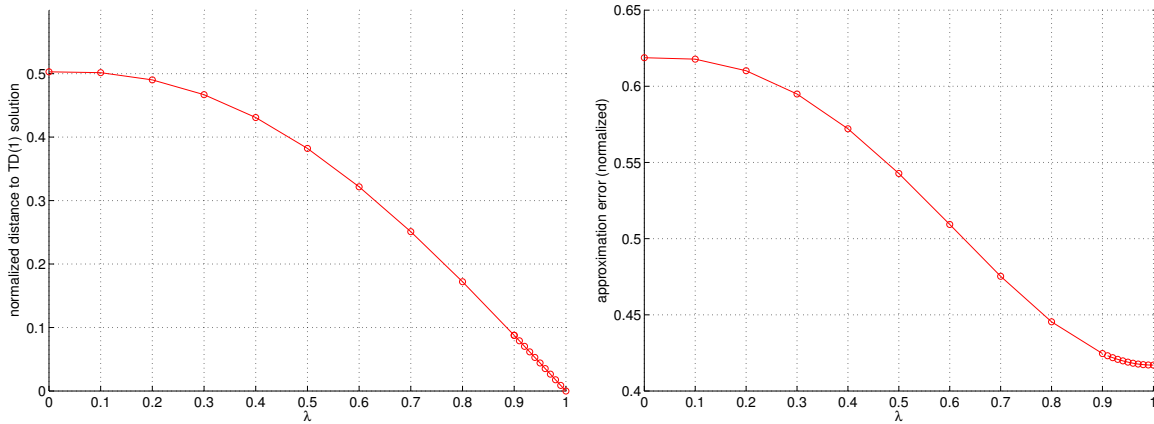


Figure 5: TD(λ) solutions for the toy problem.

Figure 5 shows the quality of the asymptotic solutions of TD(λ) with constant λ , for the full range of λ values. Plotted in the left graph is the normalized distance of the TD(λ) solution to the TD(1) solution, and plotted in the right graph is the normalized approximation error of the corresponding approximate value function, where the error is measured by the weighted Euclidean norm with weights ζ_S , and the normalization is over the weighted norm $\|v_\pi\|_{\zeta_S}$ of the true value

function v_π of the target policy. It can be seen that using $\lambda > 0.9$ provides considerably better approximations for this problem than using small λ .

We also found that for this problem if we set λ according to the Retrace algorithm with $\beta = 1$ (cf. (2.8) in Example 2.2), then the performance of LSTD is comparable to $\text{TD}(\lambda)$ with a small λ around 0.5.¹⁴ This is not surprising, because in keeping $\lambda_t \rho_{t-1} \leq 1$ always, Retrace and ABQ can be too “conservative,” resulting in an overall effect that is like using a small λ , even though λ_t may appear to be large at times. Note that this can happen to our proposed scheme too, as mentioned earlier in Remark 3.3. In the present experiment, for instance, this can happen when C is small; in particular, the case $C = 0$ reduces to $\text{LSTD}(0)$.

4.2.2 Mountain Car Problem

In this subsection we demonstrate LSTD with evolving λ on a problem adapted from the well-known Mountain Car problem [30]. The details of this adaptation, including the target and behavior policies involved, can be found in the report [38, Section 5.1, p. 23-26]; most of these details are not crucial for our experiments, so to avoid distraction, we only describe briefly the experimental setup here.

In Mountain Car, the goal is to drive an underpowered car to reach the top of a steep hill, from the bottom of a valley. A state consists of the position and velocity of the car, whose values lie in the intervals $[-1.2, 0.5]$, $[-0.07, 0.07]$, respectively. The position 0.5 corresponds to the desired hill top destination, while the position $-\pi/6$ (≈ -0.52) lies at the bottom of a valley that is between the destination and a second hill peaked at -1.2 in the opposite direction (see the illustration in Figure 6). Except for the destination state, each state has three available actions: $\{\text{back}, \text{coast}, \text{forward}\}$, and the rewards depend only on the action taken and are -1.5 , 0 and -1 for the three actions, respectively. The dynamics is as given in [30]. We consider undiscounted expected total rewards, so the discount factor is 1 except at the destination state, where the discount factor is 0 and from where the car enters a rewardless termination state permanently.

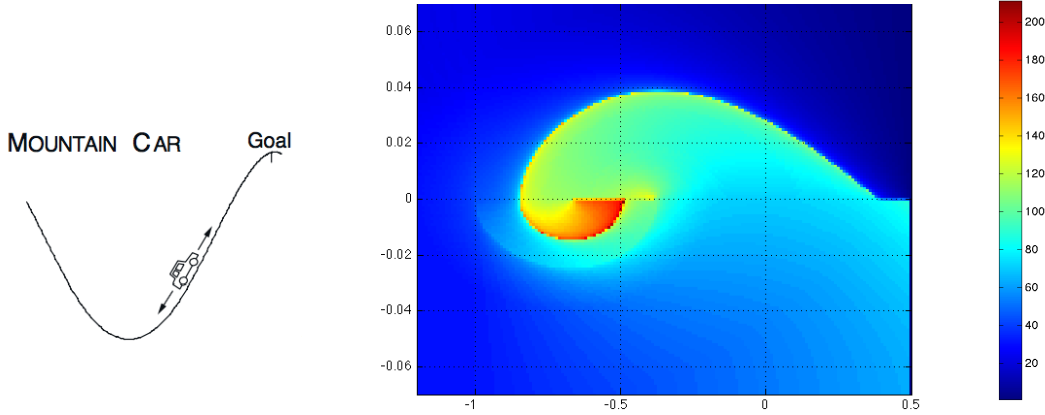


Figure 6: Costs $-v_\pi$ estimated for the Mountain Car problem (illustrated on the left).

The target policy π is a simple but reasonably well-behaved policy. On either slopes between the two hills, it tries to increase its energy (kinetic plus gravitational potential energy) by accelerating in the direction of its current motion. If this brings it up to the opposite hill (position < -1), it coasts; otherwise, if its velocity drops to near zero, it goes forward or backward with equal probability.

¹⁴Specifically, for Retrace, the normalized distance to the $\text{TD}(1)$ solution is 0.37, and the normalized approximation error is 0.54, which are comparable to the numbers for $\text{TD}(0.5)$, as Figure 5 shows.

Figure 6 visualizes the total costs $-v_\pi$ of the target policy,¹⁵ where the horizontal (vertical) axis indicates position (velocity) and the color-bar on the right shows the value corresponding to each color. The discontinuity of the function in certain regions can be seen in this figure.

The behavior policy π^o is an artificial policy that takes a random action (chosen with equal probability from the three actions) 90% of the time, and explores the state space by jumping to some random state 10% of the time. It also restarts when it is at the destination: with equal probability, it either restarts near the bottom of the valley or restarts from a random point sampled uniformly from the state space.¹⁶

We now explain how we will measure approximation qualities. Since the Mountain Car problem has a continuous state space, it is actually not covered by our analysis, which is for finite-state problems. Although we can treat it as essentially a finite-state problem (since the simulation is done with finite precision in computers), the number of states would still be too large to calculate the weights ζ_S . So, to measure weighted approximation errors $\|v - v_\pi\|$ for approximate value functions v produced by various LSTD algorithms in the subsequent experiments, we will compare v and v_π at a grid of points in the state space and calculate a weighted Euclidean distance between the function values at these grid points, using a set of weights precalculated by simulating the behavior policy.¹⁷ The image in Figure 7 (left) visualizes these weights.

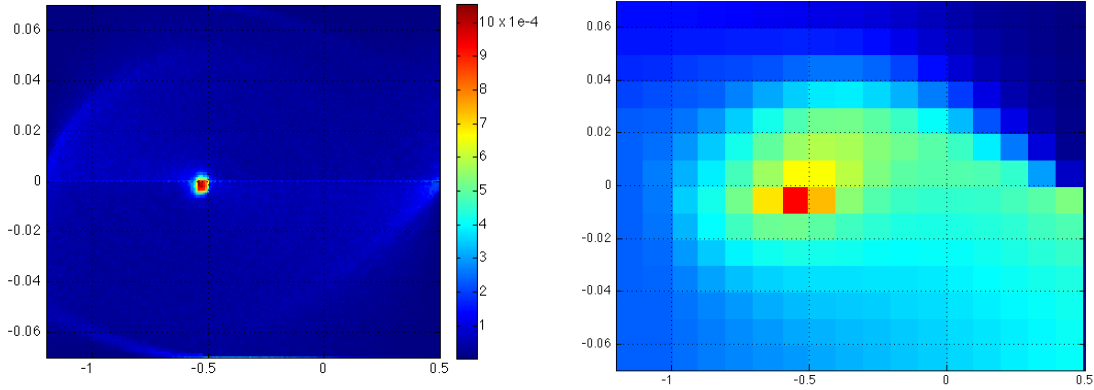


Figure 7: Left: Visualization of weights on states induced by the behavior policy. Right: approximation of $-v_\pi$ obtained from a discretized model for the Mountain Car problem.

We use tile-coding [30] to generate 145 binary features for our experiments.¹⁸ The approximate value functions obtained with LSTD algorithms are thus piecewise constant. For comparison, we also build a discrete approximate model by state aggregation. The discretization is done at a resolution comparable to our tile-coding scheme, and the dynamics and rewards of this model are calculated based on data collected under the behavior policy. The solution of the discrete approximate model is shown in Figure 7 (right) (the coloring scheme for this and the subsequent images are the same as

¹⁵The values of v_π shown in Figure 6 are estimated by simulating the target policy for each starting state in a set of 171×141 points evenly spaced in the position-velocity space. In particular, the position (velocity) interval is evenly divided into subintervals of length 0.01 (0.001), and for each starting state, the target policy is simulated 600 times.

¹⁶The behavior policy is exactly the same as described in [38, p. 24-25] except for the possibility of restarting near the bottom of the valley whenever the destination is reached.

¹⁷Specifically, we chose a grid of 171×141 points evenly spaced in the position-velocity space. We ran the behavior policy for 8×10^5 *effective* iterations, where an iteration is considered to be *ineffective* if the behavior policy takes an action, e.g., the restart action, that is impossible for the target policy. A visit to a state at an effective iteration was counted as a visit to the nearest grid point. At the end of the run, visits to a boundary point $(-1.2, 0)$ were disregarded as they were due to boundary effects in the dynamics of this problem, and the final counts were normalized to produce a set of weights on the grid points that sum to 1.

¹⁸Two tilings are used: the first (second) comprises of 64 (81) uneven-sized rectangles that cover the state space. Together they produce a total of 145 binary features. The details of the coding scheme are as described in [38, p. 28].

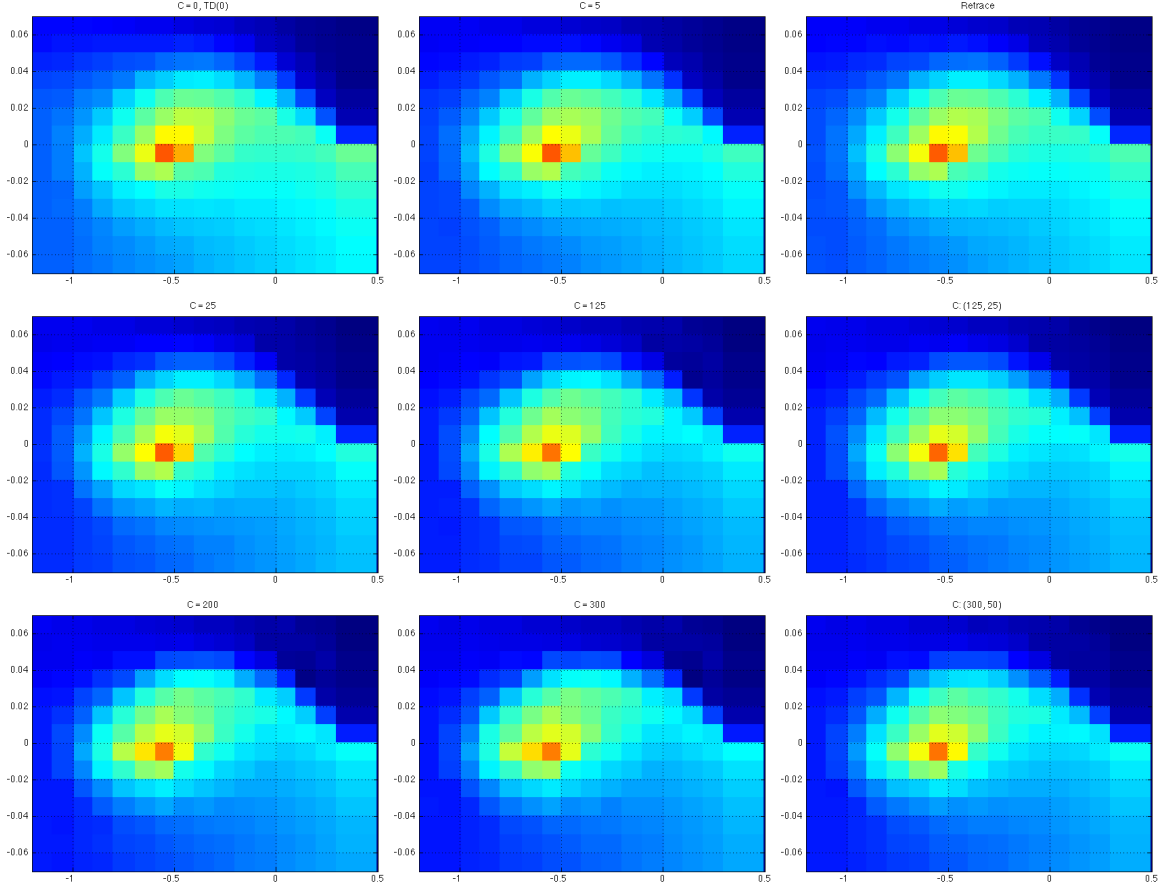


Figure 8: Approximations of $-v_\pi$ obtained by LSTD with different schemes of setting λ . Top row (from left to right): $C = 0$ (equivalent to LSTD(0)), $C = 5$, Retrace. Middle row: $C = 25$, $C = 125$, $C: (125, 25)$. Bottom row: $C = 25$, $C = 125$, $C: (300, 50)$.

shown in Figure 6). It is similar to the approximate value function calculated by LSTD(0), which is shown in Figure 8 (first image, top row).

We now report the results of our experiments on the Mountain Car problem.

First Experiment: In this experiment, we compare three ways of setting λ : (i) Retrace with $\beta = 1$ (cf. Example 2.2); (ii) our simple scaling scheme with parameter C used in the previous experiments; and (iii) a composite scheme as discussed at the end of Section 3.2, which partitions the state space into two sets¹⁹ and applies the simple scaling scheme with parameters C_1, C_2 for the first and second set, respectively. When referring to this composite scheme in the figures, we will use the designation $C: (C_1, C_2)$.

We ran LSTD with different ways of setting λ just mentioned, on the same state trajectory generated by the behavior policy, for 6×10^5 effective iterations (cf. Footnote 17). Some of the approximate value functions obtained at the end of the run are visualized as images in Figure 8. It can be seen that the result of Retrace is similar to that of the simple scaling scheme with a small C , and as we increase C , the approximation from the scaling scheme improves.

To compare more precisely the approximation errors and see how they change over time for each

¹⁹The first set consists of those states (position, velocity) with either position ≤ -0.9 or velocity ≥ 0.04 . The rest of the states belong to the second set.

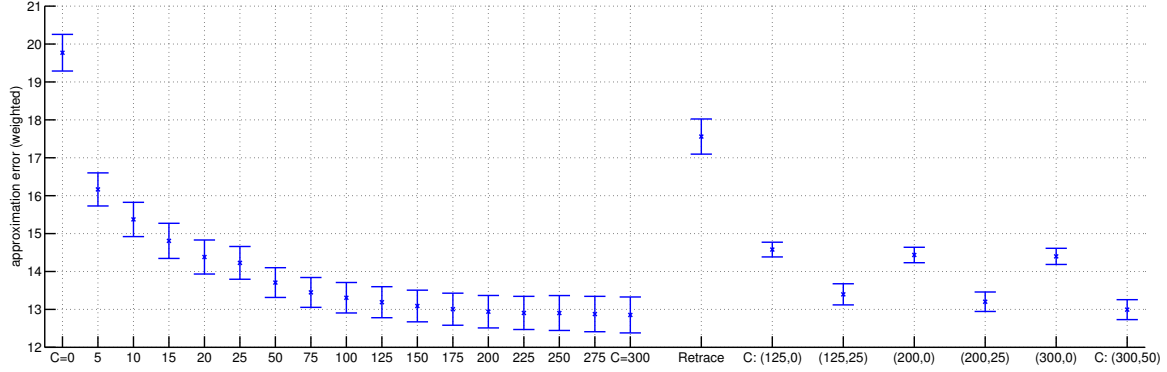


Figure 9: Compare the approximation error of LSTD for different schemes of setting λ .

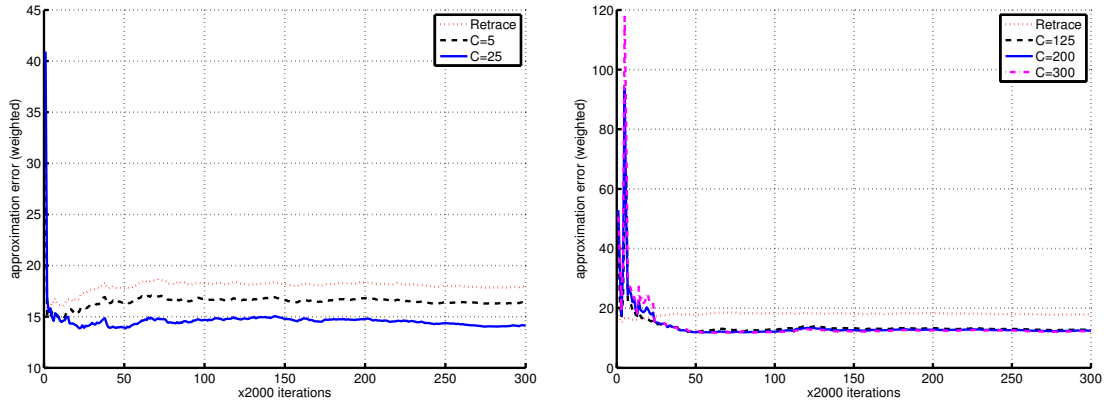


Figure 10: Compare the temporal behavior of LSTD for different schemes of setting λ .

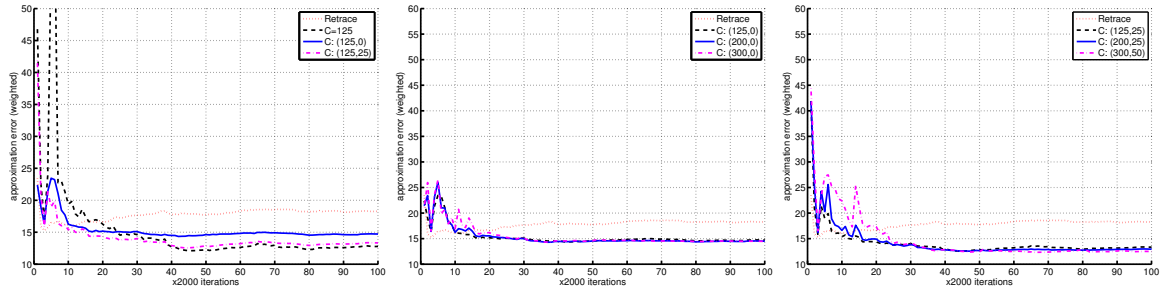


Figure 11: Compare the temporal behavior of LSTD for different schemes of setting λ .

algorithm, we did 10 independent runs, each of which consists of 6×10^5 effective iterations. The results are shown in Figures 9-11. Plotted in Figure 9 for each algorithm are the mean and standard deviation of the approximation errors for the 10 approximate value functions obtained by that algorithm at the end of the 10 runs. We can see from this figure the improvement in approximation quality as C increases. We can also see that the result of Retrace is in between those of $C = 0$ and $C = 5$, which is consistent with what the images in the top row of Figure 8 tell us.

Plotted in Figures 10-11 are the approximation errors calculated per 2000 effective iterations for each algorithm, during one of the 10 experimental runs. Figure 10 (left) shows how Retrace compares with the simple scaling with $C = 5$ and $C = 25$. It can be seen that the latter two achieved better

approximation quality than Retrace without increases in variance. Figure 10 (right) shows that for larger values of C , variances also became larger initially; however, after about 5×10^4 iterations, these schemes overtook Retrace, yielding better approximations.

Figure 11 shows how the composite scheme of setting λ performs. Comparing the plots in this figure with the right plot in Figure 10, it can be seen that the composite schemes helped in reducing variances, and in about 2×10^4 iterations the schemes $C: (125, 0)$ and $C: (125, 25)$ overtook Retrace and yielded better approximations. Together with Figure 9, Figure 11 shows clearly the bias-variance trade-off of using composite schemes in this problem.

Second Experiment: Similarly to the previous experiment, we now compare our proposed method with several other ways of setting λ for the LSTD algorithm as well as with a constrained variant of LSTD: (i) Retrace with $\beta = 1$ as before; (ii) constant λ ; (iii) constrained LSTD with constant λ ; and (iv) the simple scaling scheme with parameter C . For constant $\lambda \in [0, 1]$, the constrained LSTD(λ) used in this experiment evolves the trace vectors as off-policy LSTD(λ) does, but it forms and solves the linear equation $\frac{1}{t} \sum_{k=0}^{t-1} [e_k]_{50} \cdot \delta_k(v) = 0, v = \Phi\theta$ instead, where the function $[\cdot]_{50}$ truncates each component of the trace vector to be within the interval $[-50, 50]$.²⁰

Figure 12 is similar to Figure 9 and shows, for each algorithm, the mean and standard deviation of the approximation errors of 10 approximate value functions obtained at the end of 10 independent experimental runs (each of which consists of 6×10^5 effective iterations). The horizontal axis indicates the algorithms and their parameters. As can be seen from this figure, for constant small λ , LSTD(λ) performed well in this problem, and LSTD(0.3) and Retrace are comparable. For constant $\lambda > 0.7$, LSTD(λ) failed to give sensible results, and LSTD(0.7) started to show this unreliable behavior. Constrained LSTD(λ) is much more reliable, and it did consistently well for all values of λ tested. LSTD with evolving λ also did well, and with $C > 125$, it achieved a slightly better approximation quality than constrained LSTD(1).

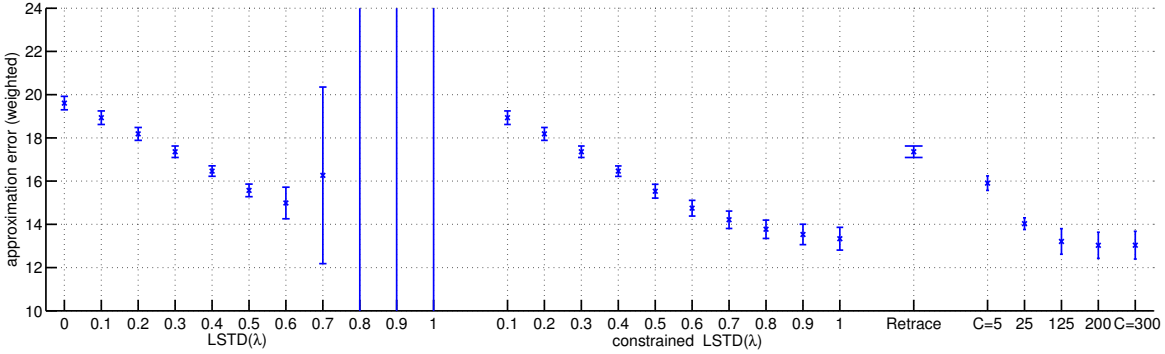


Figure 12: Compare the approximation errors of several LSTD algorithms.

Figure 13 compares the temporal behavior of several algorithms during one experimental run. Plotted are the approximation errors calculated per 2000 iterations for each algorithm in the comparison. It can be seen that in this problem constrained LSTD(λ) did not suffer from large variances even with large λ values, and compared with constrained LSTD(1), the behavior of LSTD with evolving λ was also reasonable for large C .

Third Experiment: In this experiment we first compare the simple scaling scheme and Retrace, where both schemes now use an additional parameter $\beta \in [0, 1]$, as discussed in Examples 2.1-2.2. Recall that when $\beta = 1$, they reduce to the schemes that we already compared in the previous experiments. For both schemes, as β becomes smaller, we expect the approximation quality to drop but the variance to get smaller.

²⁰Such an algorithm follows naturally from the ergodicity of the state-trace process and the approximation of an unbounded integrable function by a bounded one.

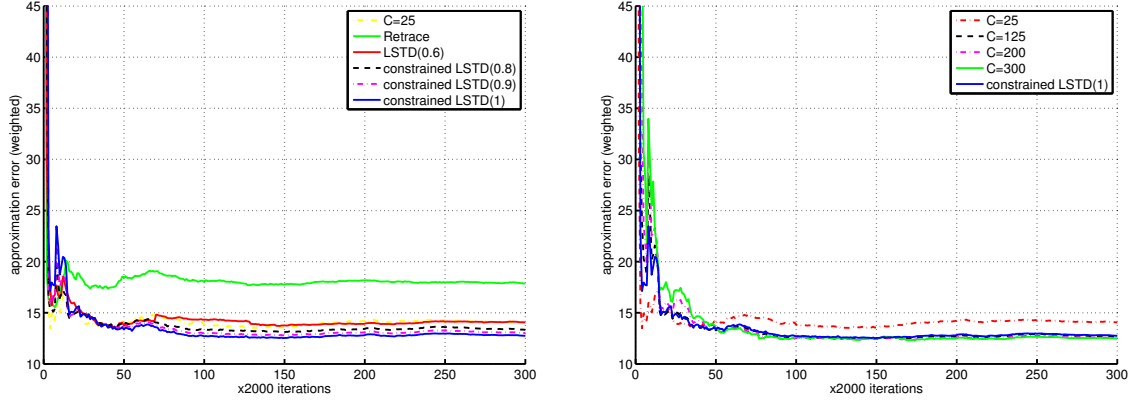


Figure 13: Compare the temporal behavior of several LSTD algorithms.

Plotted in Figure 14 for $C = 125$, $C = 200$ and Retrace are the results obtained from a single experimental run consisting of 3×10^5 effective iterations. As before the approximation errors were calculated per 2000 iterations. The results do show the expected bias-variance trade-off effects of the parameter β , during the initial period of the experimental run, although the effects on Retrace turned out to be smaller and hard to discern at the scale of the plot.

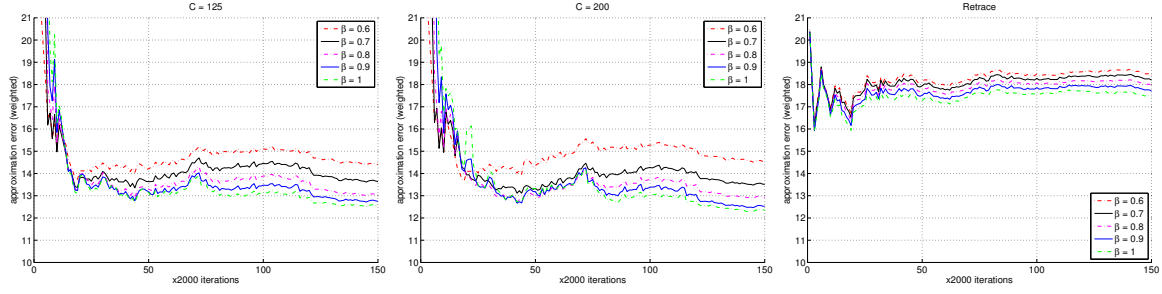


Figure 14: Compare the approximation error and temporal behavior of LSTD for different schemes of setting λ . From left to right: $C = 125$, $C = 200$, Retrace.

Next we test some of the variations on Retrace discussed in Example 2.2. Specifically, we consider (2.10)-(2.11) with parameters $\beta = 0.9$, $K \in \{1.5, 2.0, 2.5, 3.0\}$ and $C \in \{50, 125\}$. Plotted in Figure 15 are the results from one experimental run of 3×10^5 effective iterations. For comparison, the plots also show the behavior of Retrace and the simple scaling scheme with the same values of C during that run (these algorithms used the same $\beta = 0.9$). As expected and can be seen from the figure, the approximation quality improves with K and C . While the variances also tend to increase during the initial part of the run, the variants that truncate the traces by $K = 1.5$ performed comparably to Retrace initially, soon overtook Retrace and achieved better approximation quality.

5 Conclusion

We developed in this paper a new scheme of setting the λ -parameters for off-policy TD learning, using the ideas of randomized stopping times and generalized Bellman equations for MDPs. Like two recently proposed algorithms, our scheme keeps the traces bounded to reduce variances, but it is much more general and flexible. To study its theoretical properties, we analyzed the resulting state-trace process, and established convergence and solution properties for the associated LSTD

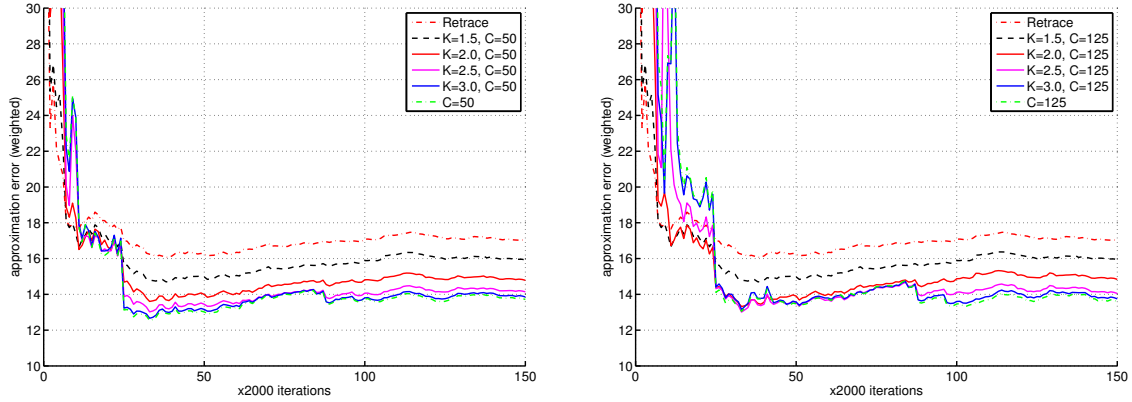


Figure 15: Compare the approximation error and temporal behavior for some variations on Retrace.

algorithm. In addition we did a preliminary numerical study. It showed that with the proposed scheme LSTD can outperform several existing off-policy LSTD algorithms. It also demonstrated that in order to achieve better bias-variance trade-offs in off-policy learning, it is helpful to have more flexibility in choosing the λ -parameters and to allow for large λ values.

Future research is to study the gradient-based implementation of the proposed scheme, and to conduct a more extensive numerical study of both least-squares based and gradient-based algorithms, with more versatile ways of using the memory states and λ -parameters, in off-policy learning applications.

Appendices

Appendix A: Proof of Theorem 3.1

We prove Theorem 3.1 in this appendix. Recall from Section 3.1 that the generalized Bellman operator T associated with a randomized stopping time τ satisfies that $v_\pi = Tv_\pi$. By (3.2), the substochastic matrix \tilde{P} in this affine operator T is given by

$$\tilde{P}_{ss'} = \mathbb{E}_s^\pi[\gamma_1^\tau \mathbb{1}(S_\tau = s')], \quad s, s' \in \mathcal{S}. \quad (\text{A.1})$$

We can extend \tilde{P} to a transition matrix \tilde{P}^e by adding an additional absorbing state Δ to the system, so that $\tilde{P}_{\Delta\Delta}^e = 1$ and

$$\tilde{P}_{s\Delta}^e = 1 - \sum_{s' \in \mathcal{S}} \tilde{P}_{ss'} = 1 - \mathbb{E}_s^\pi[\gamma_1^\tau], \quad s \in \mathcal{S}.$$

Both conclusions of the theorem will follow immediately if we show that $I - \tilde{P}$ is invertible. Indeed, if $I - \tilde{P}$ is invertible, then $v = Tv$ has a unique solution, which must be v_π since v_π is always a solution of this equation. In addition, if $I - \tilde{P}$ is invertible, then since \tilde{P} is a substochastic matrix, the spectral radius of \tilde{P} must be less than 1. Consider adding to \tilde{P} a small enough perturbation ϵM , where M is the matrix of all ones and ϵ is a sufficiently small positive number so that the spectral radius of $\tilde{P} + \epsilon M$ is less than 1. Applying [27, Theorem 1.1] to the nonnegative primitive matrix $\tilde{P} + \epsilon M$, we have that $(\tilde{P} + \epsilon M)w < w$ where w is a positive eigenvector of the matrix $\tilde{P} + \epsilon M$ corresponding to a positive eigenvalue that is strictly less than 1. Consequently $\tilde{P}w \leq (\tilde{P} + \epsilon M)w < w$, implying that \tilde{P} is a linear contraction w.r.t. a weighted sup-norm (with weights w), which is the second conclusion of the theorem.

Hence, to prove the theorem, it suffices to show that the inverse $(I - \tilde{P})^{-1}$ exists, which is equivalent to that for the Markov chain on $\mathcal{S} \cup \{\Delta\}$ with transition matrix \tilde{P}^e , all the states in \mathcal{S} are transient (see e.g., [22, Appendix A.4]).

We prove this by contradiction. Suppose it is not true, and let $\tilde{\mathcal{S}} \subset \mathcal{S}$ be a recurrent class of the Markov chain. Then for all $s \in \tilde{\mathcal{S}}$, $\tilde{P}_{ss'}^e = 0$ for $s' \notin \tilde{\mathcal{S}}$ (i.e., the submatrix of \tilde{P} corresponding to $\tilde{\mathcal{S}}$ is a transition matrix). In particular, $\tilde{P}_{s\Delta}^e = 0$, so

$$\mathbb{E}_s^\pi[\gamma_1^\tau] = 1 - \tilde{P}_{s\Delta}^e = 1, \quad \forall s \in \tilde{\mathcal{S}}, \quad (\text{A.2})$$

implying that given $S_0 \in \tilde{\mathcal{S}}$, $\gamma_1^\tau = 1$ a.s. (since $\gamma_1^\tau \in [0, 1]$). Then by (A.1),

$$\tilde{P}_{ss'} = \mathbb{E}_s^\pi[\mathbb{1}(S_\tau = s')], \quad \forall s, s' \in \tilde{\mathcal{S}}. \quad (\text{A.3})$$

Observe from (A.2) that given $S_0 \in \tilde{\mathcal{S}}$, the event $\{\tau = \infty\}$ has zero probability. This is because under Condition 2.1(i), $\gamma_1^t \xrightarrow{a.s.} 0$ (as $t \rightarrow \infty$) and consequently $\gamma_1^\infty := \prod_{t=1}^\infty \gamma_t = 0$ a.s. Indeed, the existence of the inverse $(I - P\Gamma)^{-1} = \sum_{t=0}^\infty (P\Gamma)^t$ under Condition 2.1(i) implies $(P\Gamma)^t \rightarrow 0$ as $t \rightarrow \infty$. Since the s th entry of $(P\Gamma)^t \mathbf{1}$ equals $\mathbb{E}_s^\pi[\gamma_1^t]$, we have $\mathbb{E}_s^\pi[\gamma_1^t] \rightarrow 0$ as $t \rightarrow \infty$. As the nonnegative sequence $\{\gamma_1^t\}_{t \geq 1}$ is nonincreasing (since each $\gamma_t \in [0, 1]$), this implies, by Fatou's lemma [5, Lemma 4.3.3], that $\gamma_1^t \xrightarrow{a.s.} 0$. Thus if $S_0 \in \tilde{\mathcal{S}}$, τ is almost surely finite.

We now consider a Markov chain $\{S_t\}$ with transition matrix P and $S_0 \in \tilde{\mathcal{S}}$. We will extract from it a Markov chain $\{\tilde{S}_k\}_{k \geq 0}$ with transition matrix \tilde{P}^e on the recurrent class $\tilde{\mathcal{S}}$, by employing multiple stopping times. We will show $\gamma_1^\infty = 1$ a.s., contrary to the fact $\gamma_1^\infty = 0$ a.s. just discussed.

For ease of explanation, let us imagine that there is a device that if we give it a sequence of states S_0, S_1, \dots generated according to P , it will output the stopping decision at a random time τ that is exactly the randomized stopping time τ associated with the operator T . (Because τ is a randomized stopping time for $\{S_t\}$, a device that correctly implements τ does not affect the evolution of $\{S_t\}$, so we can give $\{S_t\}$ to the device as inputs.)

Let $\{S_t\}$ start from some state $S_0 = s \in \tilde{\mathcal{S}}$. We use the device just mentioned to generate the first randomized stopping time τ_1 . We then reset the device so that it now “sees” the time-shifted process $\{S_{\tau_1+t'} \mid t' \geq 0\}$ with the initial state being S_{τ_1} . We wait till the device makes another stopping decision, and we designate that time by τ_2 . We repeat this procedure as soon as the device makes yet another stopping decision. This gives us a nondecreasing sequence $0 \leq \tau_1 \leq \tau_2 \leq \dots$.

Since $S_0 = s \in \tilde{\mathcal{S}}$, by what we proved earlier, τ_1 is almost surely finite. Let $\tilde{S}_1 = S_{\tau_1}$ and $\tilde{S}_0 = S_0$. The transition from \tilde{S}_0 to \tilde{S}_1 is according to the transition matrix \tilde{P}^e by construction (cf. (A.3)). Since $\tilde{\mathcal{S}}$ is a recurrent class for \tilde{P}^e , we must have $\tilde{S}_1 \in \tilde{\mathcal{S}}$ almost surely. Then, repeating the same argument and using induction, we have that almost surely, for all $k \geq 1$, τ_k is defined and finite and $\tilde{S}_k := S_{\tau_k} \in \tilde{\mathcal{S}}$. Thus we obtain an infinite sequence $\{\tilde{S}_k\}$, which is a recurrent Markov chain on $\tilde{\mathcal{S}}$ with its transition matrix given by the corresponding submatrix of \tilde{P}^e .

Now in view of (A.2), almost surely,

$$\gamma_1^{\tau_1} = 1, \quad \gamma_{\tau_1+1}^{\tau_2} = 1, \quad \dots, \quad \gamma_{\tau_k+1}^{\tau_{k+1}} = 1, \quad \dots \quad (\text{A.4})$$

(recall that if $\tau_k + 1 > \tau_{k+1}$, $\gamma_{\tau_k+1}^{\tau_{k+1}} = 1$ by definition). Consider first a simpler case of τ : for any initial state S_0 , $\tau \geq 1$ a.s. Then, $\{\tau_k\}$ is strictly increasing, and by multiplying the variables in (A.4) together, we have $\gamma_1^\infty = 1$ a.s. For the general case of τ assumed in the theorem, $\tau = 0$ is possible, but $\mathbf{P}^\pi(\tau \geq 1 \mid S_0 = s) > 0$ for all states $s \in \mathcal{S}$. This means that the probability of τ_k being the same for all k greater than some (random) \bar{k} is zero. So $\{\tau_k\}$ must converge to $+\infty$ almost surely, and we again obtain, by multiplying the variables in (A.4) together, that $\gamma_1^\infty = 1$ a.s. This contradicts the fact proved earlier, namely, that for any initial state S_0 , $\gamma_1^\infty = 0$ a.s. So the assumption of a recurrent class $\tilde{\mathcal{S}} \subset \mathcal{S}$ for \tilde{P}^e must be false. This proves Theorem 3.1.

Appendix B: Oblique Projection Viewpoint and Error Bound for TD

In this appendix we first explain Scherrer’s interpretation of TD solutions as oblique projections [26], and we then give approximation error bounds for TD similar to those given by [40], which do not rely on contraction properties. We will explain these properties of TD in the context of generalized Bellman operators discussed in this paper. Although this was not the framework used in [26, 40] and our setup here is more general than these previous works, the arguments and reasoning are essentially the same.

B.1 Solutions of TD as Oblique Projections of the Value Function

Let us start with the projected Bellman equation associated with TD/LSTD that we noted in Remark 3.2,

$$Tv - v \perp_{\zeta_S} \mathcal{L}_\phi, \quad v \in \mathcal{L}_\phi.$$

We can write it equivalently as

$$v = \Pi_{\zeta_S} Tv, \tag{B.1}$$

where Π_{ζ_S} denotes the projection onto the approximation subspace \mathcal{L}_ϕ with respect to the ζ_S -weighted Euclidean norm.²¹ Scherrer [26] first realized that the solution of this equation can be viewed as an *oblique projection of the value function* v_π on the approximation subspace \mathcal{L}_ϕ . This viewpoint provides an intuitive geometric interpretation of the TD solution and explains conceptually the source of its approximation bias. Analytically, this view also gives tight bounds on the approximation bias, as we will elaborate later.

An oblique projection is defined by two nonorthogonal subspaces of equal dimensions: it is the projection onto the first subspace orthogonally to the second [25], as illustrated in Figure 16. More precisely, for any two n -dimensional subspaces $\mathcal{L}_1, \mathcal{L}_2$ of \mathbb{R}^N such that no vector in \mathcal{L}_2 is orthogonal to \mathcal{L}_1 , there is an associated *oblique projection operator* $\Pi_{\mathcal{L}_1 \mathcal{L}_2} : \mathbb{R}^N \rightarrow \mathcal{L}_1$ defined by

$$\Pi_{\mathcal{L}_1 \mathcal{L}_2} x \in \mathcal{L}_1, \quad x - \Pi_{\mathcal{L}_1 \mathcal{L}_2} x \perp \mathcal{L}_2, \quad \forall x \in \mathbb{R}^N. \tag{B.2}$$

If the two subspaces are the same: $\mathcal{L}_1 = \mathcal{L}_2$ or if x lies in \mathcal{L}_1 , then the oblique projection $\Pi_{\mathcal{L}_1 \mathcal{L}_2} x$ is the same as $\Pi_{\mathcal{L}_1} x$, the orthogonal projection of x onto \mathcal{L}_1 . In general this need not be the case and $\Pi_{\mathcal{L}_1 \mathcal{L}_2} x \neq \Pi_{\mathcal{L}_1} x$ typically (cf. Figure 16). A matrix representation of the projection operator $\Pi_{\mathcal{L}_1 \mathcal{L}_2}$ is given by

$$\Pi_{\mathcal{L}_1 \mathcal{L}_2} = \Phi_1 (\Phi_2^\top \Phi_1)^{-1} \Phi_2^\top, \tag{B.3}$$

where Φ_1 and Φ_2 are $N \times n$ matrices whose columns form a basis of \mathcal{L}_1 and \mathcal{L}_2 , respectively (see [25, Chap. 1.12]). For comparison, a matrix representation of the orthogonal projection operator $\Pi_{\mathcal{L}_1}$ is $\Pi_{\mathcal{L}_1} = \Phi_1 (\Phi_1^\top \Phi_1)^{-1} \Phi_1^\top$. (Below we will use the same notation for an projection operator and its matrix representations.)

Back to the projected Bellman equation (B.1), let us assume it has a unique solution v_{TD} and express v_{TD} in terms of v_π . For the affine operator T , we have $Tv = \tilde{r}_\pi + \tilde{P}_\pi v$ for some vector \tilde{r}_π and substochastic matrix \tilde{P}_π . By Theorem 3.1, v_π is the unique solution of $v = Tv$. So $\tilde{r}_\pi = (I - \tilde{P}_\pi) v_\pi$ and

$$(I - \Pi_{\zeta_S} \tilde{P}_\pi) v_{\text{TD}} = \Pi_{\zeta_S} \tilde{r}_\pi = \Pi_{\zeta_S} (I - \tilde{P}_\pi) v_\pi. \tag{B.4}$$

Let the columns of Φ form a basis of the approximation subspace \mathcal{L}_ϕ , and let D be the diagonal matrix with ζ_S as its diagonal elements. Then a matrix representation of the projection operator

²¹In the subsequent derivations, we will not use the fact that ζ_S is the invariant probability measure induced by the behavior policy on \mathcal{S} , so the analyses we give in this appendix apply to any weighted Euclidean norm $\|\cdot\|_{\zeta_S}$.

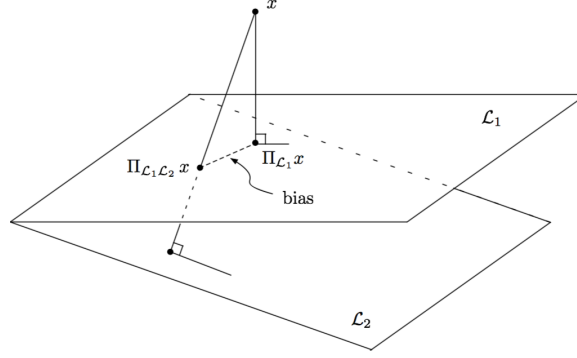


Figure 16: Oblique projection of x onto the subspace \mathcal{L}_1 orthogonally to the subspace \mathcal{L}_2 .

Π_{ζ_S} is $\Pi_{\zeta_S} = \Phi(\Phi^\top D \Phi)^{-1} \Phi^\top D$. Using this representation of Π_{ζ_S} and the fact $v_{\text{TD}} \in \mathcal{L}_\phi$, we obtain from (B.4) an expression of v_{TD} in terms of v_π :

$$v_{\text{TD}} = \Phi(\Phi^\top D(I - \tilde{P}_\pi) \Phi)^{-1} \Phi^\top D(I - \tilde{P}_\pi) v_\pi. \quad (\text{B.5})$$

(Here the invertibility of the matrix $\Phi^\top D(I - \tilde{P}_\pi) \Phi$ is equivalent to our assumption that v_{TD} is the unique solution of (B.1).)

Let us compare the expression (B.5) with (B.3). We see that if the geometry on \mathbb{R}^N is determined by the usual Euclidean norm $\|\cdot\|_2$, then v_{TD} is an oblique projection of v_π for the two subspaces $\mathcal{L}_1 = \mathcal{L}_\phi$ and $\mathcal{L}_2 = \text{column-space}((I - \tilde{P}_\pi)^\top D \Phi)$.

Alternatively, consider the case where the geometry on \mathbb{R}^N is determined by some weighted Euclidean norm $\|\cdot\|_\xi$ with weights ξ (for example, $\xi = \zeta_S$, which is one of the cases of interest in off-policy learning). In this case we can first scale each coordinate by the square root $\sqrt{\xi(s)}$ of its weight to reduce the case to that of the norm $\|\cdot\|_2$. Specifically, let \perp_ξ denote orthogonality with respect to $\|\cdot\|_\xi$, and let Ξ denote the diagonal matrix that has ξ as its diagonal elements. For the linear mapping $h : v \mapsto \Xi^{1/2} v$, we have

$$\|v\|_\xi = \|h(v)\|_2, \quad \text{and} \quad v_1 \perp_\xi v_2 \Leftrightarrow h(v_1) \perp h(v_2).$$

The second relation means that \bar{v} is an oblique projection of v for two subspaces $\mathcal{L}_1, \mathcal{L}_2$ with respect to $\|\cdot\|_\xi$ (i.e., $\bar{v} \in \mathcal{L}_1$ and $v - \bar{v} \perp_\xi \mathcal{L}_2$), if and only if $h(\bar{v})$ is an oblique projection of $h(v)$ for the two subspaces $h(\mathcal{L}_1), h(\mathcal{L}_2)$ with respect to $\|\cdot\|_2$:

$$h(\bar{v}) \in h(\mathcal{L}_1), \quad h(v) - h(\bar{v}) \perp h(\mathcal{L}_2).$$

With these facts in mind, we rewrite (B.5) equivalently as follows:

$$\begin{aligned} \Xi^{1/2} v_{\text{TD}} &= \Xi^{1/2} \Phi \cdot (\Phi^\top D(I - \tilde{P}_\pi) \Xi^{-1} \Xi^{1/2} \cdot \Xi^{1/2} \Phi)^{-1} \cdot \Phi^\top D(I - \tilde{P}_\pi) \Xi^{-1} \Xi^{1/2} \cdot \Xi^{1/2} v_\pi \\ \Rightarrow \quad h(v_{\text{TD}}) &= h(\Phi) \cdot (h(\Phi_2)^\top \cdot h(\Phi))^{-1} h(\Phi_2)^\top \cdot h(v_\pi), \end{aligned} \quad (\text{B.6})$$

where

$$\Phi_2 = \Xi^{-1}(I - \tilde{P}_\pi)^\top D \Phi \quad (\text{B.7})$$

and h applied to a matrix denotes the result of applying h to each column of that matrix. Comparing (B.6) with (B.3), we see that $h(v_{\text{TD}})$ is an oblique projection, so based on the discussion earlier, with respect to the weighted Euclidean norm $\|\cdot\|_\xi$ on \mathbb{R}^N , v_{TD} is an oblique projection of v_π for the two subspaces

$$\mathcal{L}_1 = \mathcal{L}_\phi = \text{column-space}(\Phi), \quad \mathcal{L}_2 = \text{column-space}(\Phi_2). \quad (\text{B.8})$$

Note that the second subspace \mathcal{L}_2 defining the above oblique projection is the image of the first approximation subspace \mathcal{L}_ϕ under the linear transformation $\Xi^{-1}(I - \tilde{P}_\pi)^\top D$, which *depends on the dynamics induced by the target policy as well as the generalized Bellman operator T that we choose*. Relating the oblique projection interpretation of v_{TD} to Figure 16, we can see where the approximation bias of TD, $v_{\text{TD}} - \Pi_\xi v_\pi$, comes from.

B.2 Approximation Error Bound

Let us now bound the bias term $v_{\text{TD}} - \Pi_\xi v_\pi$ relative to $\|v_\pi - \Pi_\xi v_\pi\|_\xi$, the distance between v_π and the approximation subspace measured with respect to $\|\cdot\|_\xi$. It is more transparent to derive the bound for the case of a general oblique projection operator $\Pi_{\mathcal{L}_1\mathcal{L}_2}$ with respect to the usual Euclidean norm $\|\cdot\|_2$, so let us do that first and then use the linear transformation $h(\cdot)$ to translate the result to TD as we did earlier.

We bound the bias $\|\Pi_{\mathcal{L}_1\mathcal{L}_2}x - \Pi_{\mathcal{L}_1}x\|_2$ relative to $\|x - \Pi_{\mathcal{L}_1}x\|_2$, by calculating

$$\kappa := \sup_{x \in \mathbb{R}^N} \frac{\|\Pi_{\mathcal{L}_1\mathcal{L}_2}x - \Pi_{\mathcal{L}_1}x\|_2}{\|x - \Pi_{\mathcal{L}_1}x\|_2} = \sup_{x \in \mathbb{R}^N} \frac{\|\Pi_{\mathcal{L}_1\mathcal{L}_2}(x - \Pi_{\mathcal{L}_1}x)\|_2}{\|x - \Pi_{\mathcal{L}_1}x\|_2} \quad (\text{B.9})$$

(where we treat $0/0 = 0$). This constant κ depends on the two subspaces $\mathcal{L}_1, \mathcal{L}_2$, and reflects the “angle” between them. It has several equivalent expressions, e.g.,

$$\kappa = \sup_{x \perp \mathcal{L}_1, \|x\|_2=1} \|\Pi_{\mathcal{L}_1\mathcal{L}_2}x\|_2 = \|\Pi_{\mathcal{L}_1\mathcal{L}_2}(I - \Pi_{\mathcal{L}_1})\|_2, \quad (\text{B.10})$$

or with $\sigma(F)$ denoting the spectral radius of a square matrix F ,

$$\kappa = \sqrt{\sigma(\Pi_{\mathcal{L}_1\mathcal{L}_2}(I - \Pi_{\mathcal{L}_1}) \cdot (I - \Pi_{\mathcal{L}_1})^\top \Pi_{\mathcal{L}_1}^\top)} = \sqrt{\sigma(\Pi_{\mathcal{L}_1\mathcal{L}_2}\Pi_{\mathcal{L}_2\mathcal{L}_1} - \Pi_{\mathcal{L}_1})}. \quad (\text{B.11})$$

(After the definition of κ , each expression of κ in (B.9)-(B.11) follows from the preceding one; in particular, for the last expression in (B.11), we used the fact that $\Pi_{\mathcal{L}_1}^\top = \Pi_{\mathcal{L}_1}$, $\Pi_{\mathcal{L}_1\mathcal{L}_2}^\top = \Pi_{\mathcal{L}_2\mathcal{L}_1}$, and $\Pi_{\mathcal{L}_1}\Pi_{\mathcal{L}_2\mathcal{L}_1} = \Pi_{\mathcal{L}_1}$.)

We can express κ in terms of the spectral radius of an $n \times n$ matrix, similarly to what was done in [40]. In particular, we take the last expression of κ in (B.11) and rewrite the symmetric matrix in that expression using the matrix representations of the projection operators as follows:

$$\Pi_{\mathcal{L}_1\mathcal{L}_2}\Pi_{\mathcal{L}_2\mathcal{L}_1} - \Pi_{\mathcal{L}_1} = \Phi_1(\Phi_2^\top \Phi_1)^{-1} \Phi_2^\top \cdot \Phi_2(\Phi_1^\top \Phi_2)^{-1} \Phi_1^\top - \Phi_1(\Phi_1^\top \Phi_1)^{-1} \Phi_1^\top.$$

By a result in matrix theory [8, Theorem 1.3.20], for any $N \times n$ matrix F_1 and $n \times N$ matrix F_2 , $\sigma(F_1 F_2) = \sigma(F_2 F_1)$. Applying this result to the preceding expression with $F_1 = \Phi_1$, we have that $\sigma(\Pi_{\mathcal{L}_1\mathcal{L}_2}\Pi_{\mathcal{L}_2\mathcal{L}_1} - \Pi_{\mathcal{L}_1})$ is equal to the spectral radius of the matrix

$$(\Phi_2^\top \Phi_1)^{-1}(\Phi_2^\top \Phi_2)(\Phi_1^\top \Phi_2)^{-1}(\Phi_1^\top \Phi_1) - I.$$

Combing this with (B.11), we obtain that

$$\kappa^2 = \sigma(F) - 1, \quad \text{where } F = (\Phi_2^\top \Phi_1)^{-1}(\Phi_2^\top \Phi_2)(\Phi_1^\top \Phi_2)^{-1}(\Phi_1^\top \Phi_1). \quad (\text{B.12})$$

We now translate this result to our TD context. We want to bound

$$\frac{\|v_{\text{TD}} - \Pi_\xi v_\pi\|_\xi}{\|v_\pi - \Pi_\xi v_\pi\|_\xi} = \frac{\|h(v_{\text{TD}}) - \Pi h(v_\pi)\|_2}{\|h(v_\pi) - \Pi h(v_\pi)\|_2},$$

where Π on the r.h.s. stands for the orthogonal projection onto $h(\mathcal{L}_\phi)$ with respect to $\|\cdot\|_2$. As shown by (B.6), $h(v_{\text{TD}})$ is an oblique projection of $h(v_\pi)$ for the two subspaces $h(\mathcal{L}_\phi)$ and $h(\mathcal{L}_2)$,

where \mathcal{L}_2 is given by (B.7)-(B.8). According to (B.12), the constant κ for this oblique projection is $\sqrt{\sigma(F) - 1}$ where, if we take $\Phi_1 = \Phi$ and Φ_2 as defined by (B.7), F is now given by the expression in (B.12) with $h(\Phi_1), h(\Phi_2)$ in place of Φ_1, Φ_2 , respectively. Therefore,

$$\|v_{\text{TD}} - \Pi_{\xi} v_{\pi}\|_{\xi} \leq \kappa \|v_{\pi} - \Pi_{\xi} v_{\pi}\|_{\xi} \quad \text{for } \kappa = \sqrt{\sigma(F) - 1}, \quad (\text{B.13})$$

where F is an $n \times n$ matrix given by²²

$$F = [h(\Phi_2)^{\top} h(\Phi_1)]^{-1} \cdot [h(\Phi_2)^{\top} h(\Phi_2)] \cdot [h(\Phi_1)^{\top} h(\Phi_2)]^{-1} \cdot [h(\Phi_1)^{\top} h(\Phi_1)] \quad (\text{B.14})$$

for

$$\Phi_1 = \Phi, \quad \Phi_2 = \Xi^{-1}(I - \tilde{P}_{\pi})^{\top} D\Phi. \quad (\text{B.15})$$

Note that by the definition of κ , this is a worst-case bound that depends only on the two subspaces involved in the oblique projection operator. In other words, given the approximation subspace, the dynamics \tilde{P}_{π} , the projection norm $\|\cdot\|_{\zeta_S}$ and the norm $\|\cdot\|_{\xi}$ for measuring the approximation quality, this bound is attained by a worst-case choice of the rewards \tilde{r}_{π} of the target policy. In this sense, the bound (B.13) is tight.

References

- [1] Bertsekas, D. P. and Tsitsiklis, J. N. (1996). *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA.
- [2] Borkar, V. S. (2008). *Stochastic Approximation: A Dynamic Viewpoint*. Cambridge University Press, Cambridge.
- [3] Boyan, J. A. (1999). Least-squares temporal difference learning. In *Proc. The 16th Int. Conf. Machine Learning*.
- [4] Dann, C., Neumann, G., and Peters, J. (2014). Policy evaluation with temporal differences: A survey and comparison. *Journal of Machine Learning Res.*, 15:809–883.
- [5] Dudley, R. M. (2002). *Real Analysis and Probability*. Cambridge University Press, Cambridge.
- [6] Geist, M. and Scherrer, B. (2014). Off-policy learning with eligibility traces: A survey. *Journal of Machine Learning Res.*, 15:289–333.
- [7] Glynn, P. W. and Iglehart, D. L. (1989). Importance sampling for stochastic simulations. *Management Science*, 35:1367–1392.
- [8] Horn, R. A. and Johnson, C. R. (1985). *Matrix Analysis*. Cambridge University Press, Cambridge, UK.
- [9] Karmakar, P. and Bhatnagar, S. (2015). Two timescale stochastic approximation with controlled Markov noise and off-policy temporal difference learning. arXiv:1503.09105.
- [10] Kushner, H. J. and Yin, G. G. (2003). *Stochastic Approximation and Recursive Algorithms and Applications*. Springer-Verlag, New York, 2nd edition.
- [11] Liu, B., Liu, J., Ghavamzadeh, M., Mahadevan, S., and Petrik, M. (2015). Finite-sample analysis of proximal gradient TD algorithms. In *The 31st Conf. Uncertainty in Artificial Intelligence (UAI)*.
- [12] Maei, H. R. (2011). *Gradient Temporal-Difference Learning Algorithms*. PhD thesis, University of Alberta.
- [13] Mahadevan, S., Liu, B., Thomas, P., Dabney, W., Giguere, S., Jacek, N., Gemp, I., and Liu, J. (2014). Proximal reinforcement learning: A new theory of sequential decision making in primal-dual spaces. arXiv:1405.6757.
- [14] Mahmood, A. R., van Hasselt, H., and Sutton, R. S. (2014). Weighted importance sampling for off-policy learning with linear function approximation. In *Advances in Neural Information Processing Systems (NIPS) 27*.

²²More explicitly, after substituting the expressions of $h(\Phi_1)$ and $h(\Phi_2)$ in the formula of F and removing h , we have

$$F = (\Psi^{\top} \Phi)^{-1} (\Psi^{\top} \Xi^{-1} \Psi) (\Phi^{\top} \Psi)^{-1} (\Phi^{\top} \Xi \Phi), \quad \text{where } \Psi = (I - \tilde{P}_{\pi})^{\top} D\Phi.$$

- [15] Mahmood, A. R., Yu, H., and Sutton, R. S. (2017). Multi-step off-policy learning without importance-sampling ratios. arXiv:1702.03006.
- [16] Meyn, S. (1989). Ergodic theorems for discrete time stochastic systems using a stochastic Lyapunov function. *SIAM J. Control Optim.*, 27:1409–1439.
- [17] Meyn, S. and Tweedie, R. L. (2009). *Markov Chains and Stochastic Stability*. Cambridge University Press, Cambridge, 2nd edition.
- [18] Munos, R., Stepleton, T., Harutyunyan, A., and Bellemare, M. G. (2016). Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems (NIPS)* 29.
- [19] Nummelin, E. (1984). *General Irreducible Markov Chains and Non-Negative Operators*. Cambridge University Press, Cambridge.
- [20] Precup, D., Sutton, R. S., and Dasgupta, S. (2001). Off-policy temporal-difference learning with function approximation. In *The 18th Int. Conf. Machine Learning (ICML)*.
- [21] Precup, D., Sutton, R. S., and Singh, S. (2000). Eligibility traces for off-policy policy evaluation. In *The 17th Int. Conf. Machine Learning (ICML)*.
- [22] Puterman, M. L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, New York.
- [23] Randhawa, R. S. and Juneja, S. (2004). Combining importance sampling and temporal difference control variates to simulate Markov chains. *ACM Trans. Modeling and Computer Simulation*, 14(1):1–30.
- [24] Rudin, W. (1966). *Real and Complex Analysis*. McGraw-Hill, New York.
- [25] Saad, Y. (2003). *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, 2nd edition.
- [26] Scherrer, B. (2010). Should one compute the temporal difference fix point or minimize the Bellman residual? The unified oblique projection view. In *The 27th Int. Conf. Machine Learning (ICML)*.
- [27] Seneta, E. (2006). *Non-negative Matrices and Markov Chains*. Springer, New York.
- [28] Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9–44.
- [29] Sutton, R. S. (1995). TD models: Modeling the world at a mixture of time scales. In *The 12th Int. Conf. Machine Learning (ICML)*.
- [30] Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning*. MIT Press, Cambridge, MA.
- [31] Sutton, R. S., Maei, H. R., Precup, D., Bhatnagar, S., Silver, D., Szepesvári, C., and Wiewiora, E. (2009). Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *The 26th Int. Conf. Machine Learning (ICML)*.
- [32] Sutton, R. S., Mahmood, A. R., and White, M. (2016). An emphatic approach to the problem of off-policy temporal-difference learning. *Journal of Machine Learning Res.*, 17(73):1–29.
- [33] Sutton, R. S., Szepesvári, C., and Maei, H. (2008). A convergent $O(n)$ algorithm for off-policy temporal-difference learning with linear function approximation. In *Advances in Neural Information Processing Systems (NIPS)* 21.
- [34] Tsitsiklis, J. N. and Van Roy, B. (1997). An analysis of temporal-difference learning with function approximation. *IEEE Trans. Autom. Control*, 42(5):674–690.
- [35] Ueno, T., Maeda, S., Kawanabe, M., and Ishii, S. (2011). Generalized TD learning. *Journal of Machine Learning Res.*, 12:1977–2020.
- [36] Yu, H. (2012). Least squares temporal difference methods: An analysis under general conditions. *SIAM J. Control Optim.*, 50:3310–3343.
- [37] Yu, H. (2015). On convergence of emphatic temporal-difference learning. In *The 28th Ann. Conf. Learning Theory (COLT)*. A longer version at arXiv:1506.02582.
- [38] Yu, H. (2016a). Some simulation results for emphatic temporal-difference learning algorithms. arxiv:1605.02099.
- [39] Yu, H. (2016b). Weak convergence properties of constrained emphatic temporal-difference learning with constant and slowly diminishing stepsize. *Journal of Machine Learning Res.*, 17(220):1–58.
- [40] Yu, H. and Bertsekas, D. P. (2010). Error bounds for approximations from projected linear equations. *Math. Oper. Res.*, 35(2):306–329.
- [41] Yu, H. and Bertsekas, D. P. (2012). Weighted Bellman equations and their applications in approximate dynamic programming. LIDS Technical Report 2876, MIT.